

JP07168779

Publication Title:

No title available

Abstract:

Abstract not available for JP07168779

Data supplied from the esp@cenet database - Worldwide

Courtesy of <http://v3.espacenet.com>

(19) 日本國特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-168779

(43)公開日 平成7年(1995)7月4日

(51) Int.Cl.⁶

識別記号

片内整理番号

FI

技術表示箇所

G 0 6 F 13/00

355

7368-5B

3/14

310 A

H0 4M 3/56

H0 4 N 7/15

審査請求 未請求 請求項の数 2 OL (全 17 頁)

(21)出願番号 特願平6-229274

(22)出願日 平成6年(1994)9月26日

(31)優先權主張番号 08/128,012

(32)優先日 1993年9月28日

(33)優先權主張国 米国 (U S)

(71)出願人 592089054

エイ・ティ・アンド・ティ グローバル
インフォメーション ソリューションズ イン
ターナショナル インコーポレイテッド
アメリカ合衆国 45479 オハイオ、デイ
トン サウス パターソン ブールバード
1700

(72)発明者 ジョン ヘスター
アメリカ合衆国 イリノイ州 60453 オ
ーク ローン、オースティン アヴェニュー
9212

(74)代理人 弁理士 西山 善章

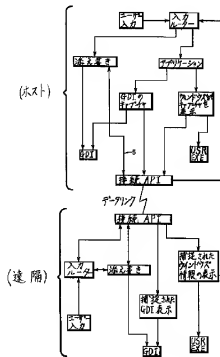
最終百に続く

(54)【発明の名称】 複数のコンピューター表示の高速複製

(57) 【要約】

【目的】 本発明は、複数コンピューターディスプレイの高速複製手段を与える。

【構成】 本発明が発生する共通の視覚的画像は、ビデオ会議に参加できずすべてのコンピュータに表示することができ、しかし本発明はその画像の全ビットマップをコンピュータに分配することはいない。その代わりに、本発明はその画像が作成されたコンピュータを分配する。さらに分配されたコマンドはデータ構造を参照する。このデータ構造はペンの色、背景の色、線幅等のパラメータを確定する。データ構造はコンピュータ内に格納されている。データ構造が変更されても、本発明はただちに情報を他のすべてのコンピュータにその変更を通知することせず、必要があるまで通知を保留する。



1

【特許請求の範囲】

【請求項1】 ホストコンピュータ上でGUIが走っており、デバイスコンテキストを参照するGDIコールを発行することにより、グラフィック画像を発生するシステムにおいて、

遠隔コンピュータに類似のデバイスコンテキストを維持し、

ホストのデバイスコンテキストに生じた変化を該遠隔コンピュータに遅滞なく送信する、ことを特徴とするコンピュータ表示の方法。

【請求項2】 ホストコンピュータ上でGUIが走っており、デバイスコンテキストを参照するGDIコールを発行することにより、グラフィック画像を発生するコンピュータシステムにおいて、

i) 選択したGDIコールを遠隔コンピュータに中継し、

ii) 該遠隔コンピュータにデバイスコンテキスト全体を送信せず、デバイスコンテキストに生じた変化を該遠隔コンピュータに通知する、ようにホストコンピュータ上でプログラムを走らせる方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明はコンピュータ会議システムに関し、特に複数のユーザーが単一のコンピュータプログラムを遠隔操作することができるシステムに関する。

【0002】

【従来の技術】 近代的電話システムは、異なる場所に居る複数のパーティーが会議を行なうことを可能にさせる。しかしながら電話会議は参加者すべてが会議室の共通のテーブルで面会する会議で得られるすべての便宜を与えることはできない。

【0003】 例えば会議室では参加者は話題に関わる図面あるいは製品等を見ることができる。その様な観察は電話会議では不可能である。

【0004】 本発明は、電話システムを使って協同ビデオ会議を行なうためのシステムに関する。本発明はいくつかのコンピュータに分配される共通の視覚的画像を発生する。当該コンピュータのユーザーはその共通画像に添え書き(annotations)をすることができる。ユーザーがこれを行なうと、本発明はすべてのコンピュータのすべてのディスプレイ上にその添え書きを複製する。

【0005】 本発明は共通の視覚的画像の発生に関して、分配されるデータ量を低減する。

【0006】

【発明が解決しようとする課題】 本発明の課題は改良された電子会議システムを与えることである。

【0007】 本発明の別の課題は、ユーザーが遠隔的にコンピュータプログラムを動作させることができるよう

2

にするシステムを与えることである。

【0008】 本発明のさらに別の課題は、複数のコンピュータの一つに存在する単一のプログラムを、それらのコンピュータが動作させることができるようにするシステムを与えることである。

【0009】 本発明のさらに別の課題は、複数のコンピュータのユーザーが共通の表示を見ることができるとともに、それに添え書きすることができるシステムを与えることである。

10 【0010】 本発明のさらに別の課題は、コンピュータ間で転送されるデータ量を低減するビデオ会議システムを与えることである。

【0011】

【課題を解決するための手段】 本発明は、共通のビデオ画像を発生するためのコマンドを送信することにより、当該ビデオ画像の複製を複数個作成する。これらコンドは、送信上、必要なデータ量がビットマップよりも小さい。

20 【0012】 しかしながら、当該画像を描くためには、コンピュータ上で走っているプログラムはコマンド自体に含まれているよりも大きなデータ量を必要とする。追加すべきデータはコンピュータに収容されているデータ構造を使って供給する。これらのデータ構造はしばしば「デバイス コンテキスト」あるいは「オブジェクト」と呼ばれ、線の色と線の幅のようなパラメータを特定する。データ構造はユーザーが変更することができ、もしもユーザーが何もしないときはデータ構造はデフォルト値に維持される。

30 【0013】 データ構造が変更されると、本発明は他のすべてのコンピュータにその変更をただちに通知することはせず、他のコンピュータがその変更について知ることが必要となるときまで待つ。

【0014】

【実施例】

概観

図1は電話線で接続された三台のコンピュータを示す。各コンピュータは、メッセージ駆動されるマルチタスクンググラフィカルユーザーインターフェイス(Graphical User Interface, GUI)を走らせる。その一例はワシントン州レッドモンド市のマイクロソフトコーポレーションから市販されているウィンドウズという名のある。そのような GUI はまたオペレーティング動作環境(operating environments)とも呼ばれる。

【0015】 GUI のユーザーはウィンドウズを使ったプログラムを介して相互作用を与える。本発明は遠方にあるコンピュータのディスプレイ全体ではなくて、選択したいくつかのウィンドウを複製する。この選択的な複製によってユーザーは自分のディスプレイ(共有されていない自分のエリア)上に個人的エリアを維持することが可能となる。各コンピュータはまた本発明者が開

3

発したソフトウェアを走らせることができる。さらに、一つのコンピュータ（ホスト）がアプリケーションプログラムを走らせる。（GUI）のマルチタスク能力があるため、ホストが 両方のプログラムを走らせることが可能である）。

【0016】本発明は4つの基本モードのオペレーションを有する。

【0017】1. アプリケーションモード (Application Mode)

図1に示す三台のコンピュータのいずれのユーザーもアプリケーションプログラムに命令を出すことができる。例えば、アプリケーションプログラムが手持ち型計算機をシミュレートするものであると仮定しよう。その初期の状態が図2に示してある。この場合、各コンピュータディスプレイが計算機を図示する。次の事象が起こると仮定する：

・ホストのユーザーが（計算機プログラムの設計に就いてキーボードの入力あるいはマウスの入力により）計算機のボタン「3」を押す。これに答えて、各計算機がその表示領域に「3」を表示する。

【0018】・遠隔使用者の一人が「+」を押す。他の遠隔ユーザーが「6」を押す。

【0019】・ホストユーザーが「=」を押す。

【0020】この時点ですべての計算機が3と6の和である「9」を表示する。ユーザーは集団でこの計算機プログラムのオペレーションを行い、各自のディスプレイがその結果を表示する。

【0021】この計算機プログラムは、合法的なシーケンスが受信される限り、いずれのユーザーがボタンを押したかは考慮せず、またユーザーの誰かがボタンを押さなかったかも考慮しない。（ただしユーザーたちは協力的であり、いずれのユーザーも計算機のオペレーションを懈怠しないと仮定する）。

【0022】2. 添書きモード (Annotation Mode)

いずれのユーザーも「ペイント」プログラムに見られるものと同様の作図ツールを使ってユーザー自身のローカルディスプレイ上に描画することができる。ユーザーは、ボックス、円、円弧、楕円等を作図することができる。ユーザーはまたディスプレイ上の項目を消去することができる。

【0023】本発明はユーザー達の添書きを他のすべてのディスプレイ上に複製することができ、したがってすべてのユーザーが類似のディスプレイを見ることができ。しかし次のような因子があるので、それらの表示は異なることがあり得る。

【0024】(A) ディスプレーモニターが異なれば、色の表現能力等の性質が異なる。

【0025】(B) ディスプレーのプロトコル (IGA, BGA など) が異なればグラフィックイメージの表現が異なり、表示能力が異なる。

4

【0026】(C) GUIが異なり、あるいは同一の GUI でもバージョンの異なる場合は表示方式が異なることがある。

【0027】(D) ユーザーの何人かは計算機が表示されるウィンドウの大きさを変更することがあり、その結果縮尺に差異が生じていることがある。

【0028】これらの差異は表示されるイメージの外観に差を生じさせる。しかしすべてのディスプレイの基本的な内容は同一なはずである。すなわち差異を許容するために、本発明は適当な縮尺で作図をすることができ。

【0029】3. ローカル添書きモード (Local Annotation Mode)

ユーザーはローカルディスプレイに添書きをすることができるが、それらの添書きは秘密に保たれるので、他のユーザーはその添書きを見ることができない。

【0030】4. 観察モード (View Mode)

いずれのユーザーも添書きをすることができず、命令を発行することもできない。しかし添書きに類似した行動を取ることができ。ユーザーは自分たちのカーソルを移動させることができ、他のユーザーはその動きを見ることができるので、遠隔的指摘（カーソルによる表示物の位置の指摘）ができる。観察モードは、例えば添書きモードの実行中にある特定ユーザーのモードを観察モードに指定する実施例などによって有用である。この実施例ではすべてのユーザーが添書きをすることができるが「観察」ユーザーは見ることができただけであり、添書きをすることができない。

【0031】各モードの説明

図3-図14は例として計算機プログラムの使用を通して色々なモードを図示する。

【0032】図3

ホストがアプリケーションプログラムを走らせる。

【0033】モードは「添書き」である。

【0034】ユーザーの入力はホストコンピュータで行なわれる。

【0035】ユーザーが計算機のオペレーションを試みる。

【0036】ホストコンピュータのユーザーが計算機を使って二つの数の加算を試みると仮定する。その試みの対象となる最初の数について考える。

【0037】ホストユーザーはホストのカーソルを計算機のキー上に移動させ、マウスをクリックすることによりキー入力を試みる。しかしマウスのクリックはアプリケーションプログラムに届かない。なぜならば本発明がそれを阻止するからである。アプリケーションプログラムは応答しない。なぜならばプログラムがマウスのクリックを受理しないからである。

【0038】すなわち、さらに詳細に説明すると、GUIはこのマウスの移動を検出し、発生すべき「マウスメッセージ」を発生させる。GUIはこのマウスメッセー

5

ジをキュー (queue) におく。マウスメッセージはこのキューの中で処理を待機する。図15に示す「入力ルーター (INPUT ROUTER)」がこれらメッセージを読む。

「添え書きモード」が現在実行中であるので、入力ルーターはこのメッセージを添え書きブロックに送る。アプリケーションはこのメッセージを受理せず、従って応答しない。このマウスクリックは無視される。

【0039】「添え書き」の応答

「添え書き」はマウスのメッセージに対して2通り(あるいはそれ以上)の応答をするように構成することができ、その一つの構成では、添え書きはマウスが初めに添え書きツールを選択することを要求する。選択がま

たたく行われないと、「添え書き」はマウスメッセージを無視する。
【0040】選択は通常「ペイント」プログラムで行なわれるようにツールイメージ上でマウスをクリックすることにより行なう。「添え書き」はこのツールの選択を認識し、次いでそれ以降のマウスのクリックは選択ツールで描画するためのデータとして扱う。例えば長方形ツールが選択されたときは、次の二つのマウスのクリックが長方形の対角線上の隅を決定する。(後で議論する図4が長方形の描画を示す。)

第二の構成では、添え書きモードにあるときはペンのようなデフォルトツールが自動的に選択される。この構成ではユーザーが(クリックすることにより)計算機のボタンを押す試みを行うと、そのペンを使ったユーザーの線の描画(意図せず)に始まる。これを認識したユーザーは公知方法のいずれかを使って線の描画を修正させることができる。

【0041】それゆえ添え書きモードでは本発明は(a)デフォルトの「添え書き」を開始することによってマウスの入力に応答する、あるいは(b)「添え書き」ツールが選択されないためにマウスの入力を無視する、のいずれかの応答をする。ユーザーからのキーボード入力も同様に扱われる。もちろん「添え書き」が他の応答をするように設計することもできる。

【0042】カーソルの追跡
各ディスプレイは関連のマウスで位置決定できるカーソルを示す。本発明はすべてのディスプレイ上に各カーソルを複製することができる。この様にして図3に示すように3つマウスを使うとき、3つのカーソルがある。(ただし簡単のため、ただ一つのみ図示してある。)

その結果、一人のユーザーがマウスを動かすとそれに対応するカーソルがすべてのディスプレイ上で運動する。

【0043】一般的に3つのカーソルは識別が可能である: 各カーソルは固有の色・形状・ラベルの導入等によりその所有者を区別する。

【0044】図4

ホストがアプリケーションプログラムを走らせる。

【0045】モードは添え書きモードである。

6

【0046】ユーザーの入力はホストコンピュータで行なう。

【0047】ユーザーは計算機の上にボックスの描画を試みる。

【0048】この状況は図3に示すものと極めて類似している。ただし今の場合ユーザーが図3においてはボタンを押すことを意図している代わりに添え書きを描画しようと試みている点と異なる。

【0049】ホストコンピュータのユーザーが計算機の上にボックスを描画すると仮定する。(このボックスは強調のため過度に大きく図示されている。ボックスは計算機自身を越えて伸びないことが望ましい。)本発明は遠隔のコンピュータ上にそのボックスを複製する。

(このボックスは図示していない添え書きツールを使って描かれる。)

【0050】図15で見て、入力ルーターは論理の流れを「添え書き」宛てに送る。「添え書き」はそのボックスを描くための適当なGDI機能を呼び出す。また「添え書き」は「添え書きメッセージ」を接続APIに送る。この接続APIが遠隔者に添え書きメッセージを届ける。

【0051】図15に示す「添え書き」はこの添え書きメッセージを受信する。この「添え書き」ブロックは各遠隔コンピュータにおいて実行される論理を表す。この「添え書き」はブロックGDIを経由して適当なGDI機能を呼び出す。

【0052】「GDI」とはグラフィカル デバイス インターフェイスの略号である。「GDI機能」はGDI、EXEと呼ばれる大きなプログラムに含まれている。GDI機能は、呼び出されると、その後でユーザーが入力する入力に基づいて円、ボックスあるいはテキストのような特定のグラフィックイメージを描く。ほかのGDI機能はペン幅の選択などの他のタスクを行なう。

【0053】GDI EXEは市販されている製品である。GDI、EXEに関する技術的な詳細はマイクロソフト社から入手可能な「ウィンドウズ ソフトウェア開発キット」およびチャールズ ベルトン著の「ウィンドウズ図3.1のプログラミング」(ワシントン州レッドモンド市: マイクロソフト プレス、ISBN 1-55615-395-3、1992年発行)に記載されている。

【0054】図5

ホストがアプリケーションプログラムを走らせる。

【0055】モードは「アプリケーション」である。

【0056】ユーザー入力はホストコンピュータで行なわれる。

【0057】ユーザーは計算機の使用を試みる。

【0058】ホストユーザーは計算機のキー「3」の上にカーソルを移動させマウスをクリックする。GUIはマウスメッセージを発生し、それをキューにおく。本発明はマウスメッセージを呼んで、そのメッセージをアプ

7

リケーションプログラム（すなわち計算機プログラム）に送る。このプログラムは、(1) キー「3」が押されたこと、及び(2) GDIのコールを使って計算機のディスプレイに数字「3」を描画する、という応答をする。

【0059】このアプリケーションプログラムはまた、そのプログラムの内部的オペレーションを行うため、ユーザーが「3」を入力したという事実を記録する。

【0060】本発明は、計算機の中に「3」を描画している際、および押されている「3」のボタンを表示する際に、アプリケーションプログラムが出すGDIコールを中断する。他のコンピュータは同一のGDI機能を実行することによりホストディスプレイを複製する。このGDI中断に関する一層詳細な説明は「一般的な考察」に題する後の節で述べる。

【0061】この様にして、ホストのユーザーが計算機を動作させることを全てのユーザーが同時に見ることができる。（きわめて短い遅延が関与するので、このホストの動作は厳密に同時的ではない。とは言え、ホスト及び遠隔のコンピュータが並んで動作したとしても人は多分その遅延に気付くことができないであろう。）図15においてマウスメッセージがアプリケーションプログラムに指向されるべきであるということを入力ルーターが認識し、論理の流れをアプリケーション（すなわち計算機プログラム）に指向する。アプリケーションは、

(1) 押された「3」のキーを描画し、(2) 適当なGDI機能呼び出す。ことによって計算機のディスプレイ内に数字「3」を書き込む。

【0062】しかし本発明はそれらが実行される前に、アプリケーションプログラムのGDI呼び出しを図15に示すGDIキャプチャー経由で捕捉（キャプチャー）する。本発明は捕捉した呼び出し（コール）に付いてふたつのことを行なう。その一つとして、本発明は「接続API」ブロックを経由してこれらの呼び出しを他のコンピュータに通告する。このアクションは図15に示す「捕捉されたGDIディスプレイ」ブロックに属く。そうするとブロックGDIによって示すようにこのブロックが各遠隔者に同一のGDI機能を実行させる。

【0063】その二として、図15に示すブロックGDI経由で本発明は、ホストで実行すべきアプリケーションプログラムが呼び出すGDI機能をホストで実行することを許す。

【0064】それゆえ、本発明はアプリケーションプログラムが行なったGDI機能の呼び出し（コール）を捕捉する。本発明は捕捉したコールを遠隔コンピュータに通告する結果、遠隔コンピュータはそれらのコールを複製することができる。本発明は、その捕捉したコールが意図されたようにホスト上で実行されることを許す。

【0065】図6

8

ホストがアプリケーションプログラムを走らせる。

【0066】モードは「ローカルな添え書き」である。

【0067】ユーザー入力力はホストコンピュータで行なわれる。

【0068】ユーザーが計算機に演算を行なわせることを試みる。

【0069】添え書きモードにおいて、ユーザーに与えられているデフォルト添え書きツールがまったく無いと仮定しよう。この仮定の下で、ユーザーがカーソルを計算機のボタンまで移動させてボタンを「押す」と、図15の「入力ルーター」がマウスメッセージを「添え書き」ブロックまで通過させる。このマウスクリックは有効な添え書き入力シーケンスではない（ツールが何も選択されていない）ので、何の「添え書き」も描かない。

【0070】更に、遠隔コンピュータは、ホストコンピュータのマウスに対応するカーソルの移動を示さない。なぜならば前述したように、「ローカルな添え書き」の実行中は図15のライン5が添え書きメッセージを他のコンピュータに送らないからである。

【0071】更にまた、ホストディスプレイ上で計算機のボタンを押しても、それに応答して押されたボタンを描き直すことはしない。なぜならば、アプリケーションがマウスメッセージを受信しなかったからである。アプリケーションが責任をもつのは押された計算機ボタンの描画である。

【0072】添え書きのデフォルトがローカル添え書きモードをとっているユーザーに指定されているときは、そのユーザーのマウスクリックによってそのツールによる描画が開始される。ユーザーが誤りに気づいたとき、ユーザーは公知の方法で描画を終了するだろう。

【0073】図7

ホストがアプリケーションプログラムを走らせる。

【0074】モードは「ローカルな添え書き」である。

【0075】ユーザー入力力はホストコンピュータで行なわれる。

【0076】ユーザーは計算機に添え書きを試みる。

【0077】これらの条件のもとで、図15のA部に示す入力ルーターはボックスの描画などの有効な添え書きを認識する。入力ルーターはその論理の流れを「添え書き」ブロックに指向し、「添え書き」ブロックが図7に示すボックスの描画等の添え書きのための適当なGDI機能呼び出す。

【0078】しかし、添え書きがローカルであるので、図7に示すように遠隔コンピュータ上には何のボックスも描かれない。図15のデータ路5に沿って何のデータも送られない。

【0079】図8

ホストがアプリケーションプログラムを走らせる。

【0080】モードは「観察」である。

9

【0081】ユーザー入力はホストコンピュータで行う。

【0082】ユーザーが計算機の実操作を試みる。

【0083】図8が示すようにマウスのクリックは無視され、遠隔コンピュータには何も起きない。

【0084】図15において入力ルーターはこのマウスメッセージを読むが、アプリケーションに届かないようにそれを阻止する。なぜなら現在のモードが「観察モード (View Mode)」だからである。

【0085】図9

ホストがアプリケーションプログラムを走らせる。

【0086】モードは「添え書き」である。

【0087】ユーザー入力は遠隔コンピュータで行なう。

【0088】ユーザーが計算機の実操作を試みる。

【0089】ユーザーがマウスカーソルを計算機のボタン上に移動させ、マウスをクリックすると仮定する。このクリックは無視される。他のコンピュータ (ホスト及び他の遠隔コンピュータ) はユーザーのカーソルの運動を示すが、それ以外は何も示さない。その理由は、ツールが何も選択されていないからである。

【0090】図15において、入力ルーターはこのマウスメッセージがアプリケーションに到達することを阻止する。論理演算は「添え書き」に当てられ、GDIブロックを経由して「添え書き」がユーザーの遠隔ディスプレイ上にカーソルを描画する。「添え書き」もまた接続APIにデータを送り、この接続APIが論理演算を図15の「添え書き」に当てる。この「添え書き」は、他の二つのコンピュータ (すなわちホスト及び当該他の遠隔コンピュータ) 上に存在する添え書き論理内容を表す。これらの「添え書き」ブロックはそれぞれのユーザーのカーソルに対応するカーソルを、図15に示すGDIブロックを経由してそれぞれの場所で描画する。図15のGDIはGDI機能コールの内容を表わす。

【0091】ホストがボックス描画ツールのような一つのツールを使うことができる一方、遠隔コンピュータは円を描くツールのような別のツールを使用することができ。

【0092】図10

ホストがアプリケーションプログラムを走らせる。

【0093】モードは「添え書き」である。

【0094】ユーザー入力は遠隔コンピュータで行なう。

【0095】ユーザーが計算機への添え書きを試みる。

【0096】添え書きはボックスであると仮定する。全てのディスプレイ上にボックスが描かれる。図15において、ユーザーの遠隔コンピュータにある入力ルーターがマウスメッセージを「添え書き」ブロックに送る。

10

「添え書き」は二つのことを行なう。その一つは「添え書き」が添え書きであるボックスの描画を行なうための適当なGDIを呼び出す。

【0097】その二は、「添え書き」が接続APIに添え書きメッセージを送り、接続APIが他方のコンピュータに添え書きメッセージを送る。ただしこれらの一方はホストコンピュータであり、他方は遠隔コンピュータである。ホストの論理プロセスが図15の「添え書き」に到達し、前記他方の遠隔コンピュータの論理プロセスが図15にある「添え書き」に到達する。

【0098】これらの「添え書き」ブロックは両者とも、適当なGDI機能呼び出しでこれにユーザー所望の添え書きを描かせる。しかしホストの論理経路5はこの時点で取り上げられない。なぜなら、ホストの添え書きを他のコンピュータに複製する必要がないからである。

【0099】図11

ホストがアプリケーションプログラムを走らせる。

【0100】モードはアプリケーションである。

【0101】ユーザー入力は遠隔コンピュータで行なう。

【0102】ユーザーが計算機の実操作を試みる。

【0103】読者はこの計算機プログラムがホストにのみロードされていることを思い起こされたい。ただし遠隔コンピュータのユーザーがそれを動作させたいと希望している。

【0104】図15に示す遠隔ユーザーの入力ルーターは、マウスメッセージを接続APIに送る。ホストはこれらのメッセージを受信し、これらのメッセージは図15に示すホストの入力ルーターに届けられる。ホストの入力ルーターはこれらメッセージをブロックアプリケーション (すなわち計算機プログラムであるアプリケーションプログラム) に送る。このプログラムは二つの重要なことを行なう。

【0105】遠隔マウスがそれらメッセージ出したにも関わらず、計算機プログラムはそれらのメッセージをあたかもホストのマウスによって発行されたかのように扱う。計算機プログラムはユーザーがデータ (すなわち「3」) を入力したことを知ると、通常の方法で応答する。その応答は次のことを含んでいる：(1) 押された計算機ボタン「3」を示すこと。(2) 計算機のディスプレイ内に数字「3」を書き込むこと。及び(3) それ自身の内部的計算を行なうこと。

【0106】しかし、計算機プログラムが前節の(1)

(2) を実行することができるようになるまでに、本発明は最初に計算機プログラムが呼び出すGDI機能捕捉する。この捕捉は図15のGDI捕捉ブロックに例示されている。この捕捉の際、本発明は二つのことを行なう。その一つは、(他のコンピュータが使用できるよ

うにするため)本発明がこれらのGDI機能を接続APIに送る。ユーザーの遠隔コンピュータにおいては、図15に示す接続APIがこのGDI機能を「捕捉されたGDIディスプレイ」に送る。このディスプレイはホストディスプレイの複製を作成する。その二として本発明は(図15に示すブロックGDIを経由して)ホストで実行されるべきGDI機能をホストで実行させる。それゆえ、事象の一般的シーケンスは次の通りである:

・遠隔ユーザーが計算機ボタンを押す試みをする。

【0107】・遠隔コンピュータ上で動作している本発明は、この試みを検出し、ホスト上で走っている計算機プログラムにデータを送る。このデータはメッセージの形式を取る。計算機のプログラムはこのメッセージがホストのマウスからきたものと「考える」。

【0108】・計算機プログラムが通常通りに働き、GDIコールを経由してホストディスプレイ上にイメージを描く。

【0109】・本発明がGDIコールを捕捉し、それらを遠隔コンピュータに通報する。

【0110】・遠隔コンピュータはホストのウィンドウを複製する。遠隔ユーザーはこのようにして、ホスト上で走っている計算機プログラムを遠隔的に動作させることができる。

【0111】別の言い方で要約すると、本発明は遠隔コンピュータのマウスメッセージに基づいてマウスメッセージをホストに発生する。(ホスト上で走っている)計算機プログラムが、それらのマウスメッセージがあたかもホストで発生されたかのようにマウスメッセージに応答する。本発明は計算機プログラムが行ったGDIコールを中断し、遠隔コンピュータにおける同一のGDIコールを実行することにより、ホストディスプレイを遠隔コンピュータに複製する。

【0112】図12

ホストがアプリケーションプログラムを走らせる。

【0113】モードは「ローカルな添え書き」である。

【0114】ユーザー入力は遠隔コンピュータで行なう。

【0115】ユーザーが計算機の実操作を試みる。

【0116】当該ユーザーのマウスクリックは無視される。ツールの選択に失敗しているため、このマウスの運動に反応して他のディスプレイ上に何も現われない。

【0117】図13

ホストがアプリケーションプログラムを走らせる。

【0118】モードは「ローカルな添え書き」である。

【0119】ユーザー入力は遠隔コンピュータで行なわれる。

【0120】ユーザーが計算機への添え書きを試みる。

【0121】上に記したように、添え書きはユーザーのディスプレイ上に描かれる。他のディスプレイ上には何

の添え書きも起きない。

【0122】図14

ホストがアプリケーションプログラムを走らせる。

【0123】モードは「観察」である。

【0124】ユーザー入力は遠隔コンピュータで行なう。

【0125】ユーザーが計算機の実操作を試みる。

【0126】ユーザーのディスプレイ上でマウスのカーソルが移動するが、マウスクリックは無視される。また他の二つのディスプレイはユーザーのマウスカーソルの運動を示さない。

【0127】一般考察

1. 別異のプログラムはディスプレイ全体のうちの異なる部分で描画を行う。ディスプレイはGDI機能を使って描かれる。しかし一つのディスプレイでも異なる部分はそれぞれ別のプログラムによって描かれる。

【0128】これらすべての描画オペレーションがGDI機能を使用して行なわれるにも拘らず、GDI機能はコンピュータ間の排他的ディスプレイ複製通信媒体ではない。

【0129】添え書きは一つの形態のコンピュータ間データ転送を含む。

【0130】アプリケーションプログラムによる描画は他の形態のものを含む。

【0131】例えば、ユーザーが添え書きを行なうとき、ユーザーのマウスメッセージは図15に示す経路5を経由して、他のコンピュータにおいて「メッセージ」として複製される。これらの複製メッセージは次いで、(他のコンピュータにおける)それぞれの「添え書き」ブロックがその添え書きを描くための適当なGDIコールを発行させる。すなわちGDIコールは、他の添え書きを行なっているユーザーからは直接に他のコンピュータに送られない。

【0132】それとは対照的に、アプリケーションプログラムがグラフィックイメージをディスプレイ上に描かせている場合、本発明はそのGDIコールを(図15に示す「GDI捕捉」を経由して)中断し、そのGDIコールを他のコンピュータ上に複製させる。

【0133】相違の理由

二つの異なる手順(マウスメッセージの複製及びGDIコールの複製)を行なう主な理由は、添え書きがディスプレイ情報とは異なったロケーションに格納されるためである。

【0134】すなわち図2の計算機に戻って、アプリケーションプログラムは次のような一般的な方法で計算機のイメージを格納する。添え書きデータは本発明によって格納される。アプリケーションプログラムデータは(ホストにある)アプリケーションプログラムによって格納される。各キーのイメージはGDI機能がそのキー

13

を引き出すことのできるデータとして格納される。このデータは位置、大きさ、色等の情報を含む。各キーは関連の数を含む。その数は位置・大きさ・フォントタイプなどの情報とともにテキストキャラクターとして格納することができる。

【0135】添え書きデータは異なった場所に格納されるが同じ一般的な方法で格納される。

【0136】添え書きあるいはアプリケーションプログラムのいづれかがビットマップを必要とするときは公知の方法でビットマップがGUIにより格納される。

【0137】本発明は、マスキングに関する公知の技術によって添え書きイメージをアプリケーションのイメージと組み合わせる。すなわち本発明は、遠隔コンピュータが受信したGDI機能をビットマップに再生する。本発明は、受信した添え書き情報を、それとは異なるビットマップに再生する。これら二つのビットマップは相互にマスク掛けされる。

【0138】ユーザーがアプリケーションイメージを添え書き抜きで保存できるようにする等の目的で、添え書きデータはアプリケーションデータから分離して保存される。この代わりとして、ユーザーは添え書きデータを単独に、あるいは添え書き済み表示の形で、保存することができる。

【0139】別個保存するもう一つの理由は、添え書きデータを別個に保存することは添え書きデータを全く有しないディスプレイを描くのに便利であるからである。添え書きデータが計算機イメージデータと混同してしまったときは、添え書きデータを除去することは不可能でないまでも困難である。

【0140】もしもGDIコールを排他的に送信するとすると(すなわちメッセージの複製が全く行なわれないとすると)、添え書きデータを別個に格納するように構成するための余分の作業が必要となる。

【0141】2. GD I の中断もしくは捕捉

GD I の中断は次のように理解することができる。

【0142】A. スタートアップの時点で、本発明は各GD I 機能の最初の5バイトを、ある特定のプログラムすなわち Trap. GDI へのジャンプ指令(JUMP)で置換する。

【0143】B. Trap. GDI は、希望のグラフィックイメージに対するパラメータを獲得する(例えばボックスの場合はふたつの対角線の角の位置など)。そしてサブプログラムであるPkgDispCallを呼び出す。Trap. GDI はまた最初の5バイトをも置換する。

【0144】C. PkgDispCallは、Trap. GDI からパラメータを受信してオブジェクト構造体(object structure)を発生する。このオブジェクト構造体はほかのコンピュータがボックスを描くのに必要な全ての要素を含んだデータブロックである。

【0145】例えば、オブジェクト構造体はボックスの

14

寸法及び位置に関する情報を含んでいる。更にGUIは、「コンテキスト(context)」内にイメージを描く。このコンテキストは、ペンの巾・色そのほかの特徴的なものを含んでいる。本発明は各コンピュータのコンテキストを追跡する。もしも描いたボックスのコンテキストが遠隔コンピュータのコンテキストと異なっていると、他のコンピュータが正しいコンテキストを発生するのに必要なデータをPkgDispCall が含める。

【0146】D. オブジェクト構造体は他のコンピュータに発送され、当該他のコンピュータが同一のGDI機能を実行する。

【0147】E. 本発明はオリジナルのGDI機能を実行する。

【0148】3. ディスプレイ全体が転送されることはない。ディスプレイはビット単位で置換されることはない。例えば、図2に示す計算機のイメージはビット単位でコンピュータ間を転送することもできる。もしも計算機が200×300ピクセルの空間を占めるならば、6万ピクセル(すなわち200×300のピクセル)に関する情報が送らなければならない。この代わりとして、図2に示す特定の計算機は18個の長方形および18個の長方形各々に対する一テキスト文字として扱われ、その結果全部で34個のオブジェクトとなる。各オブジェクトは、大きさ及び位置のようなパラメータを必要とする。パラメータの数は3から3乃至10である。10個のパラメータがあると仮定すると、340個のデータを送らなければならない。もちろん、各データピース(データ片)の大きさは色々の因子に依存するが、各ピースには小さなバイト数を想定しては足りる。

【0149】それゆえ、本発明はビット単位の複製では6万ピース必要なデータを、オブジェクト複製のための最大340ピースに低減する。もちろんオブジェクトによってはビットマップの形態を取ることがあり、ビットごとに送らなければならない。しかし一般的にビットマップは稀にしか現れない。更に、一般的にビットマップを送るときは一回だけ送られると予想される。

【0150】さらにオブジェクトデータは可能であれば圧縮される。すなわちコンピュータ間の情報伝達は、可能であれば、圧縮したデータで行なわれる。圧縮は本技術分野で公知である。

【0151】4. データリンクの形態
コンピュータ間の通信はいくつかの形態を取り得る。市販されているネットワーク、ローカルエリア、及びワイドエリアを使うことができる。市販されているISDN電話回線は地方の電話会社から提供されており、これらを使うこともできる。モデム通信を使うこともできる。

【0152】5. 先行技術のメッセージ検出
入力デバイスにตอบสนองしてGUIで発生したメッセージを検出する市販のパッケージが利用可能である。そのようなバ

15

ッページの一つは WINSIGHT であり、これはポーランドインターナショナルから入手できる。しかしこれらのパッケージは遠隔コンピュータにメッセージ通報しないとされる。

【0153】6. 代わりとなる「GDI 捕捉」上記のグラフィックの捕捉に代わる方法は次の通りである。本システムが与える GDI を、実際のシステム GDI を呼び出す前に GDI コールを処理する別個のプロシージャで置き換える。システム GDI の名前は二つのモジュール間で混同することを防止するために変更される。システムに提供されているモジュールを介して行なわれる GDI コールを捕捉するために、USR.EXE にも同じテクニックが使われる。

【0154】7. 一つ以上のコンピュータがアプリケーションプログラムを走らせることができる。一つのプログラムに対してある特定のコンピュータがホストとして行動することができ、別のプログラムに対して遠隔コンピュータが行動することができる。例えば一つのコンピュータはワープロを走らせることができ、別のコンピュータは描画プログラムを走らせることができる。それぞれのプログラムに対しては各々がホストである。

【0155】各コンピュータに搭載された本発明のソフトウェアは同一もしくは実質上同一であるので、すべてのユーザーがワードプロセッサまたは CAD プログラムを上述したように走らせることができる。

【0156】8. 「リアル」カーソルおよび「疑似」カーソル二つのタイプの「カーソル」がある。各 GUI はそれ自身の「リアル (real, 実際の)」カーソルを所有する。実際のカーソルは GUI 機能によっては発生されず、GUI 中の独立した機能によって発生される。このカーソルは、マウスの運動に応答して GUI が移動させるビットマップと見做すことができる。ローカルマウスによって制御されるそのリアルカーソルに加えて、本発明は各遠隔の参加者に対する「疑似」カーソルを発生する。疑似カーソルは GDI 機能を使って発生される。時折リアルカーソルはカーソルの移動とともに変化する。例えば、ツールバー上にあるときはカーソルは矢印の形状を取ることができ、クライアント領域上にあるときは手の形に変化することができる。しばしばこの変化はアプリケーションプログラムの制御のもとにおかれる。

【0157】それゆえ、(図 11 に示すように) 遠隔ユーザーがホストマシン上で走っているアプリケーションプログラムを制御するときは、GDI コールを使用せずにアプリケーションプログラムがホストマシン上のカーソルを変化させることができる。従って、図 15A 及び 15A に示す GDI の捕捉は、変化したカーソルを遠隔ディスプレイ上で複製することに対して非効率である。

16

【0158】この問題に対処するため、本発明はリアルカーソル (例えば SetCursor コマンド等) を変化させる機能を監視する。本発明は遠隔コンピュータ上でカーソル変化を複製する。

【0159】同一の SetCursor コマンドを実行することでも一つの方法である。もう一つの方法は、適当な GDI コールのシーケンスを実行することによって、ホストカーソルが変化するとき遠隔カーソルを変化させ、またはビットマップを描かせることであろう。

【0160】9. ディスプレイ全体は複製されない。本発明はディスプレイのユーザーが同定するウィンドウを複製するだけである。すなわちユーザーはノートパッドのような作図を会議の期間中秘密に保持することができる。GDI コールはタスクハンドル (task handle) を使用する。共用するアイテムをタスクハンドルが参照しないならば、GDI コールは共用されない。

【0161】10. GDI コールのすべてのパラメータは必ずしも常に共用しない。

【0162】本発明のこの特徴はおそらく、以下の「背景」に基づいて最も良く示される。

【0163】背景 GDI コールが発行されると、二つの主要な特徴が喚起される。それらは「デバイスコンテキスト」と「オブジェクト」である。最初にデバイスコンテキストを考える。

【0164】デバイスコンテキスト

黒板上に一つの長方形を描くことと、その長方形は黒の背景上に描かれたある幅の白線から成るであろう。この状況は、黒の背景上に設けたある線幅の白線からなる型の「デバイスコンテキスト」を例示している。黒板上に (同じチョークを使って) 描いたような画像も、このデバイスコンテキストに符合する。

【0165】ビデオディスプレイ、プリンター、あるいはプロッターのようなコンピュータの出力デバイス上に長方形を描くため、GUI を使用するとき、同様の考えが当てはまる。我々は自分の描きたいデバイスコンテキストを明示的に、あるいはデフォルトによって、特定する。

【0166】GUI が使用するデバイスコンテキストには、次のパラメーターが含まれる：

- 長方形の色、
- 背景の色、
- 長方形を構成する四本の線の太さ、
- 長方形を収容するウィンドウの大きさと位置、および
- その他のデータ。

【0167】(好ましくは本実施例の GUI は約 19 個の異なるパラメーターを収容するデバイスコンテキストを使用する。)

オブジェクト

オブジェクトにも類似の概念が当てはまる。長方形を描

17

くためのGDIを発行するときは「ペン」を選択する。このペンは物理的なペンではなく、描くべき線を記述するパラメータの集合である。このペンのパラメータは線の幅、色およびダッシング（点描）の型（実線、破線、中心線等）である。これらのパラメータはコンピュータにより格納されており、ペンをアクションに呼び込むときに使用される。

【0168】このようにしてGUIは複数のソースから描画指令を得る。ここで考えている例では、以下の三つのソースがある：

(1) GDIコール。これは画像の型（長方形）その他大きさおよび位置のようなパラメータを特定する。
 【0169】(2) デバイスコンテキスト。これは（デフォルトによってではなく）GDIコールによって特定される。

【0170】(3) 関与するオブジェクト。これも（デフォルトによってではなく）GDIコールによって特定される。

【0171】本発明が遠隔コンピュータにGDIコールを中継（relay）するときは、遠隔コンピュータは関与するデバイスコンテキストおよびオブジェクトが何であるか通知を受けなければならない。本発明は、所定帯域幅を確保するため、送信する情報を圧縮する。

【0172】例

次のことを仮定しよう。

【0173】(1) 遠隔会議セッションがちょうど今開始した。

【0174】(2) アプリケーションが共用されている。

【0175】(3) 今より前にGDIコールは発行されていない。

【0176】(4) デバイスコンテキストもペンも、デフォルト状態から修正されていない。

【0177】(5) APP_WINDOW という名前のウィンドウ内に、その共用されているアプリケーションが現われている。

【0178】(6) ホストコンピュータにいるユーザーが「APP_WINDOW」上に円を描くことを希望している。

【0179】添え書きをする者がマウスを使って円を描くと、本発明は上述したようにマウスメッセージを受信し、適当なGDIコールを発行する。GDIコールはデバイスコンテキストおよびペンの両方について明示的にあるいはデフォルトにより、参照を行ない、その結果GUIが円を描くことができるようになる。仮定により、デバイスコンテキストおよびペンのデフォルト値には何ら変更がなされていないので、GUIは円を描くのにデフォルト値を使用する。

【0180】当該アプリケーションが共用されているため、本発明はGUIを呼び出す途上でGDIコールに割

18

り込み、描かれた円のことを遠隔コンピュータに通知する。本発明は、(a) GDIコールの同一性と共に、

(b) デバイスコンテキスト、および(c) ペンに関する情報を遠隔コンピュータに通知しなければならない。このとき本発明に重要でない他のデータも併せて通知される。

【0181】デバイスコンテキストおよびペンのデフォルト値に何ら変更がなされていないので、本発明は実際、デフォルトデバイスコンテキストおよびデフォルトペンの両方を使用するよう、遠隔コンピュータに通告する。本発明はデフォルトデバイスコンテキスト全体を送信することはせず、デフォルトペン全体を送信することもしない。

【0182】他の例

添え書きをする者がデバイスコンテキストまたはペンのいずれかを変更する、と仮定しよう。例えば、添え書き者は背景の色を変更し（従ってデバイスコンテキストを変更する）、あるいはペンの幅を変更する（それゆえペンのパラメータを変更する）ことができる。

【0183】本発明は、遠隔コンピュータにデバイスコンテキストの全体を送信することはせず、ペンパラメータの集合全体を送信することもしない。その代わりに本発明は、当該変更のみを送信する。

【0184】さらに、本発明は、変更されたデバイスコンテキストあるいは変更されたペンの参照を行なうGDIコールが発行されたときの、当該変更のみを送信する。この遅延が重要となる場合の一例を示そう。

もう一つの例

遠隔会議セッションの期間中に、ある添え書き者が添え書きをする準備をしていて、ウィンドウの背景の色を変更し、したがって当該ウィンドウのデバイスコンテキストを変更する、と仮定する。その変更直後、ある会議参加者がある声明を行なうと仮定する。この声明が添え書き者の注意を引き、このため、行なわれようとしている目前の添え書きが中断される。

【0185】本発明は二つの主な理由からこの時点における当該デバイスコンテキストの変更を送信しない。第一に、差し迫った必要がない。というのは、デバイスコンテキストを必要とするGDIコールが発行されていないからである。第二に、そのような送信は帯域を占有することになるが、それよりも他のトランザクション（処理）のために使用したほうがよい。この例では上記声明は大量のデータ送信の開始を合図することができる。

【0186】その代わり、添え書き者が、変更したデバイスコンテキストを使用するGDIコールを発行させるまで、本発明は待機する。その時期が来たとき、その時点で本発明は遠隔者に変更を中継する。しかしながら、再び繰り返して言うが、この時点でもデバイスコンテキスト全体は送信されない。送信されるのは変更だけである。

19

【0187】さらに別の例

遠隔コンピュータは、当該GDIコールに関わるデバイスコンテキストおよびオブジェクトの両方を格納している。遠隔コンピュータが変更を受信すると、遠隔者は格納されているデバイスコンテキストおよびオブジェクトを適宜修正する。従って遠隔コンピュータはホストコンピュータに格納されている同一のデバイスコンテキストおよびオブジェクトに対応するデバイスコンテキストおよびオブジェクトを維持する。ホストは遠隔者と共に必要に応じて、当該デバイスコンテキストおよびオブジェクトを更新する。全く更新されないときは、デバイスコンテキストおよびオブジェクトはそれらのデフォルト値を維持する。

【0188】遠隔者がそのデバイスコンテキストを使用せず、ホストのGDIを使用する描画

ホストコンピュータが遠隔者にGDIを送ると、遠隔コンピュータは自分対応する自身のデバイスコンテキストおよびオブジェクト中にすでに収容されている情報を使って、GDIコールによりリクエストされている画像を描く。すなわち、ホストは指定されたコールに必要なデバイスコンテキスト全体もオブジェクト全体も送信しない。

【0189】本発明のこの局面は、既に進行している会議に遠隔コンピュータが参加するときに、特に意味がある。ホストコンピュータは使用中のデバイスコンテキストおよびオブジェクトの現在のステータスを維持している。しかし、新規に参加する遠隔コンピュータは現在のステータスを知らない。

【0190】それにもかかわらず、本発明はその遠隔者に情報を送信するので、その情報から遠隔者はデバイスコンテキストおよびオブジェクトを構築することができる。要するにこの情報は、生じた変更である。遠隔者は、既知のデフォルトの内容および当該変更に基づいて、デバイスコンテキストおよびオブジェクトを構築する。

【0191】上記の議論は添え書きについて考えた。しかし、本発明は添え書きを行わない遠隔コンピュータに対し、GDIコールを発行し、それを送信する。これらの例では、本発明は上述したように動作する：遠隔者はホストから受信した変更に基づいて、デバイスコンテキストおよびオブジェクトを維持する。

【0192】定義

「デバイスコンテキスト」は、描く画像の背景を特定する変更可能なデータの集合、と定義することができる。そのようなデータの集合の例が、前記バットソルドの本の514ないし515頁の表に記載されている。

【0193】ホストのデバイスコンテキストに生じた変更に基づいて、かつ当該デバイスコンテキスト全体を送信することなく、遠隔コンピュータ内にデバイスコンテキストを維持する方法は、「非冗長」維持法（“non-r

20

edantant” maintenance）と呼称することができる。

【0194】すなわち、ホストのデバイスコンテキストが19個のパラメータを受容しており、単一のパラメータが変更されたと仮定しよう。変更後に19個のパラメータすべてを遠隔者に送信することは冗長であろう。遠隔者は既に18個のパラメータ値を知っているからである。

【0195】注意点

好ましい本実施例ではペンのようなオブジェクトに対して変更が行なわれると、当該変更だけでなくオブジェクト全体が他のコンピュータに送信される。その理由は、一般にオブジェクト全体が占めるデータ空間が小さいからである。

【0196】

【発明の効果】本発明の改良された電子会議システムは、ユーザーが遠隔的にコンピュータプログラムを起動させることができるため、複数のコンピュータの一つに存在する単一のプログラムを起動させることができる。

【0197】加えて、本発明では複数のコンピュータのユーザーが共通の表示共用することができる。すなわち、共通の表示に添え書きすることができる。遠隔コンピュータへのかかるデータ通信は変更にかかるデバイスコンテキストおよびオブジェクト情報送信すればよいので、少量のデータ通信で済み、またデータは圧縮して送ることができるので、高速通信できるビデオ会議システムを与えることができる。

【図面の簡単な説明】

【図1】 電話線で接続された三台のコンピュータを例示する図である。

【図2】 図の3ないし図14に詳細に説明する実施例の一例を示す。本例では計算機プログラムがいろいろのパーティーにより作動され、添え書きを付けられる。

【図3】 本発明が添え書きモードにあるときにホストユーザーの一人が計算機のオペレーションを試みる場合、本発明がどのように応答するかを示す図である。

【図4】 本発明が添え書きモードにあるときにホストユーザーの一人が計算機に添え書きを試みる場合、本発明がどのように応答するかを示す図である。

【図5】 本発明がアプリケーションモードのあるときにホストユーザーの一人が計算機のオペレーションを試みる場合、本発明がどのように応答するかを示す図である。

【図6】 本発明がローカル添え書きモードにあるときにホストユーザーの一人が計算機のオペレーションを試みる場合、本発明がどのように応答するかを示す図である。

【図7】 本発明がローカル添え書きモードにあるときにホストユーザーの一人が計算機に添え書きを試みる場合、本発明がどのように応答するかを示す図である。

【図8】 本発明がビューモードにあるときにホストユ

21

一ザ一にどのように応答するかを示す図である。

【図 9】 本発明が添え書きモードにあるときに遠隔のユーザーが計算機の実行を試みる場合、本発明がどのように応答するかを示す図である。

【図 10】 本発明が添え書きモードにあるときに遠隔のユーザーが計算機に添え書きを試みる場合、本発明がどのように応答するかを示す図である。

【図 11】 本発明がアプリケーションモードにあるときに遠隔のユーザーが計算機の実行を試みる場合、本発明がどのように応答するかを示す図である。

10

22

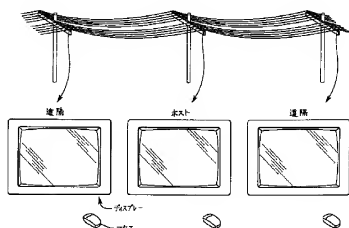
【図 12】 本発明がローカル添え書きモードにあるときに遠隔のユーザーが計算機の実行を試みる場合、本発明がどのように応答するかを示す図である。

【図 13】 本発明がローカル添え書きモードにあるときに遠隔のユーザーが計算機に添え書きを試みる場合、本発明がどのように応答するかを示す図である。

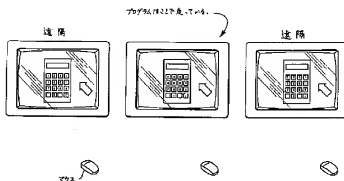
【図 14】 本発明がビューモードにあるときに遠隔のユーザーにどのように応答するかを示す図である。

【図 15】 本発明で使用する論理の流れを例示する図である。

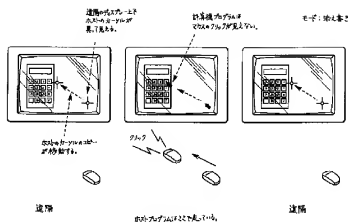
【図 1】



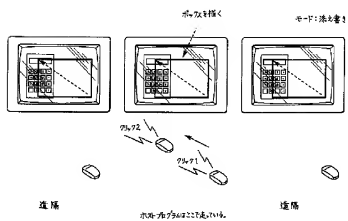
【図 2】



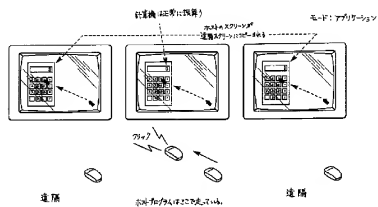
【図3】



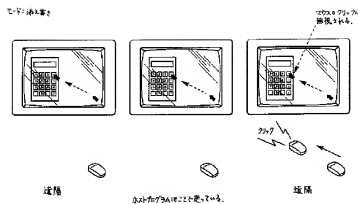
【図4】



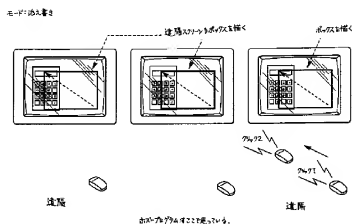
【図5】



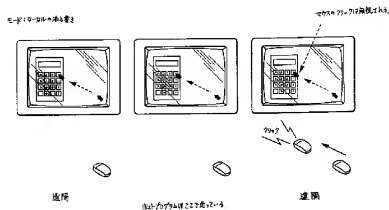
【図9】



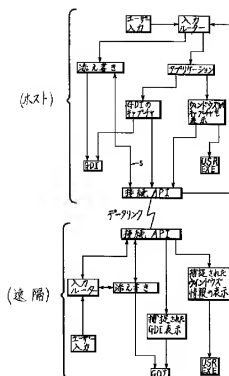
【図10】



【図12】



【図15】



フロントページの続き

(72)発明者 クリスタ エス、シュワルツ
 アメリカ合衆国 イリノイ州 60510 バ
 タビア、ジェニファー ドライブ 953

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-168779

(43)Date of publication of application : 04.07.1995

(51)Int.Cl.

G06F 13/00

G06F 3/14

H04M 3/56

H04N 7/15

(21)Application number : 06-229274

(71)Applicant : AT & T GLOBAL INF SOLUTIONS
INTERNATL INC

(22)Date of filing : 26.09.1994

(72)Inventor : HESTER JOHN
SCHWARTZ KRISTA S

(30)Priority

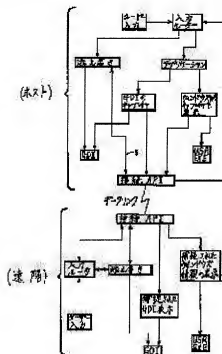
Priority number : 93 128012 Priority date : 28.09.1993 Priority country : US

(54) HIGH-SPEED COPY FOR DISPLAY ON PLURAL COMPUTERS

(57)Abstract:

PURPOSE: To provide a high-speed copy means for plural computer displays.

CONSTITUTION: A common visual image to be generated can be displayed on all the computers which can be participated in a video conference. However, all the bit maps of that image are not distributed to the computers. In place of that, a command for preparing that image is distributed. The distributed command refers to data structure. This data structure determines parameters such as the color of pen, the color of background and the width of line. The data structure is stored in the respective computers. Even when the data structure is changed, the information of that change is not immediately reported to all the other computers but the report is held until required.



TRANSLATION OF JAPANESE PATENT NO. JP07-168779

CLAIMS

[Claim 1] By publishing a GDI call which GUI is running on a host computer and refers to a device context, A method of a computer display characterized by what a device context similar to a remote computer is maintained, and change produced in a host's device context is transmitted for to this remote computer without delay in a system which generates graphic images.

[Claim 2] By publishing a GDI call which GUI is running on a host computer and refers to a device context, In a electronic computing system which generates graphic images, a GDI call of which: i) selection was done is relayed to a remote computer, ii) A method of running a program on a host computer like which does not transmit the whole device context to this remote computer, but notifies change produced in the device context to this remote computer.

DETAILED DESCRIPTION

[0001] [Industrial Application] This invention relates to the system which can operate a computer program with two or more single users by remote control especially about a computer conference system.

[0002] [Description of the Prior Art] A modernistic phone system enables two or more parties which are in a different place to hold a conference. However, the conference call can give no facilities obtained at the meeting in which all participants meet on the common table of a conference room.

[0003] For example, in a conference room, the participant can see a drawing or a product in connection with subject, etc. Such observation is impossible in a conference call.

[0004] This invention relates to the system for performing common video conferencing using a phone system. This invention generates the common visual image distributed to some computers. The user of the computer concerned can put a postscript to the common picture (annotations). If a user performs this, this invention will reproduce the postscript on all the displays of all the computers.

[0005] This invention reduces the data volume distributed about generating of a common visual image.

[0006] [Problem(s) to be Solved by the Invention] The technical problem of this invention is giving the improved electronic meeting system.

[0007] Another technical problem of this invention is giving the system by which it enables it for a user to operate a computer program remotely.

[0008] Another technical problem of this invention is giving the system by which it enables it for those computers to operate the single program which exists in one of two or more of the computers.

[0009] Another technical problem of this invention is giving the system to which it can be added while the user of two or more computers can see a common display.

[0010] Another technical problem of this invention is giving the video conference system which reduces the data volume transmitted between computers.

[0011] [Means for Solving the Problem] This invention creates two or more duplicates of the video image concerned by transmitting a command for generating a common video image. Required data volume of these commands is smaller than a bit map on transmission.

[0012] However, in order to draw the picture concerned, a program which is running on a computer needs big data rather than contained in the command itself. Data which should be added is supplied using a data structure accommodated in a computer. These data structures are often called a "device context" or an "object", and specify a parameter like a color of a line, or width of a line. A user can change a data structure, and a data structure is maintained by default value when a user does nothing.

[0013] If a data structure is changed, this invention does not carry out notifying the change to other computers of all the immediately, but it will wait for it until it is necessary for other computers to get to know about the change.

[0014] [Example] General-view drawing 1 shows three computers connected with the telephone wire. Each computer runs the multitasking graphical user interface (Graphical User Interface, GUI) by which a message drive is carried out. The example has a thing named the Windows marketed from Microsoft Corporation of Redmond, Washington. It is such. GUI is also called operating operating environment (operating environments) again.

[0015] The user of GUI gives an interaction via the program using Windows -- it can do. This invention reproduces not the whole display of the computer which exists far away but the window of shoes to choose. A user becomes possible [maintaining individual area on its own display (one's area which is not shared)] by this alternative duplicate. Each computer can run the software which this invention person developed again. One computer (host) runs an application program. (Since it has the multitasking ability of GUI, a host it is possible to run both programs).

[0016] This invention has the operation of four dominant modes.

[0017] 1. Application mode (Application Mode) Any user of three computers shown in drawing 1 can give a command to an application program. For example, an application program will assume that it is what simulates a stock type computer. The first state is

shown in drawing 2. In this case, each computer display illustrates a computer. The user of : and the host who assumes that the following phenomenon happens pushes the button "3" of a computer (it responds to the design of a computer program and they are an input of a keyboard, or an input of a mouse). This is answered and each computer displays "3" on the viewing area.

[0018] One of remote users pushes "+." Other remote users push "6."

[0019] A host user pushes "=".

[0020] At this time, all the computers display "9" which is the sum of 3 and 6. Users perform operation of this computer program in a group, and each one of displays display that result.

[0021] This computer program does not take into consideration whether it did not take into consideration whether which user pushed the button, and someone of users pushed the button, as long as a lawful sequence is received. (However, it is assumed that users are cooperative and neither of the users does nonfeasance of the operation of a computer).

[0022] 2. Postscript mode (Annotation Mode) Any user can draw on a user's own local display using the same drawing tool as what is seen by the "paint" program. The user can draw a box, a circle, a circle, an ellipse, etc. The user can also eliminate the item on a display.

[0023] This invention can reproduce users' postscript on other displays of all the, therefore can see a display with all the similar users. However, since there are the following factors, those displays may differ.

[0024] (A) If display monitors differ, character, such as expression ability of a color, differs.

[0025] (B) If the protocols (IGA, BGA, etc.) of a display differ, expressions of a graphical image differ and display ability differs.

[0026] (C) GUI differs. Or it is the same. When GUI also differs in a version, display types may differ.

[0027] (D) Some users might change the size of the window where a computer is displayed, and, as a result, the difference may have produced them in the contraction scale.

[0028] These differences make the appearance of the image displayed produce a difference. However, the fundamental contents of all the displays are the same ****. Since the difference in a size is permitted, this invention can be drawn by suitable contraction scale.

[0029] 3. Local postscript mode (Local Annotation Mode) Although the user can put a postscript to a local display, since those postscripts are kept secret, other users cannot see the postscript.

[0030] 4. Observation mode (View Mode) Neither of the users can put a postscript and a command cannot be published. However, action similar to a postscript can be taken. Since the user can move his cursor and other users can see the motion, remote indication (indication of the position of a display thing with cursor) can be performed. Observation mode is useful in the example etc. which specify the mode of the specific user who is, for example during execution in postscript mode as observation mode. In this example, although all the users can put a postscript, "observation" user can only see and cannot put a postscript.

[0031] Explanatory view 3 in each mode - drawing 14 illustrates various modes through use of a computer program as an example.

[0032] The drawing 3 host runs an application program.

[0033] The mode is a "postscript."

[0034] A user's input is performed with a host computer.

[0035] A user tries the operation of a computer.

[0036] It is assumed that the user of a host computer tries addition of two numbers using a computer. The first number that is the target of the trial is considered.

[0037] A host user moves a host's cursor on the key of a computer, and a keystroke is tried by clicking a mouse. However, the click of a mouse does not reach an application program. It is because this invention prevents it. An application program does not answer. It is because a program does not receive the click of a mouse.

[0038] That is, if it explains still in detail, GUI will detect movement of this mouse and will generate the "mouse message" which should be generated. GUI sets this mouse message at cue (queue). A mouse message stands by processing in this cue. "The input router (INPUT ROUTER)" shown in drawing 15 reads these messages. Since "postscript mode" is performing now, an input router sends this message to a postscript block. Application does not receive this message, therefore does not answer. This mouse click is disregarded.

[0039] The response "postscript" of a "postscript" can be constituted so that two kinds (or more than it) of responses may be carried out to the message of a mouse. In the one composition, a postscript requires that a mouse should choose a postscript tool first. If selection is not performed at all, a "postscript" will disregard a mouse message.

[0040] Selection is performed by clicking a mouse on a tool image so that it may usually be carried out by a "paint" program. A "postscript" recognizes selection of this tool and, subsequently treats the click of the mouse after it as data for drawing with a selection tool. For example, when a rectangle tool is chosen, the click of the following two mice becomes final and conclusive the corner on a rectangular diagonal line. (Drawing 4 about which it argues later illustrates rectangular drawing.)

In the second composition, when it is in postscript mode, a default tool like a pen is chosen automatically. With this composition, if a user performs the trial which pushes the button of a computer (it clicks), drawing of the line of the user using that pen will start (** which is not meant). The user who has recognized this can make drawing of a line finish using either of the publicly known methods.

[0041] So, postscript In the mode, since the (b "postscript") tool is not answered or chosen as the input of a mouse by starting the "postscript" of the (a) default, this invention carries out the response of ***** which disregards the input of a mouse. The keyboard input from a user is treated similarly. It can also design so that a "postscript" may carry out other responses, of course.

[0042] Pursuit each display of cursor shows the cursor which can be spotted with the mouse of relation. This invention can reproduce each cursor on all the displays. Thus, as shown in drawing 3, when using 3 mice, there is three cursor. (However, since it is easy, even free accepts it and it has illustrated it.)

As a result, if a one user moves a mouse, the cursor corresponding to it will exercise on all the displays.

[0043] Generally :each cursor which three cursor can identify identifies the owner by introducing for example, a color, shape, and a label etc.

[0044] The drawing 4 host runs an application program.

[0045] The mode is postscript mode.

[0046] A user's input is performed with a host computer.

[0047] A user tries drawing of a box on a computer.

[0048] This situation is extremely similar with what is shown in drawing 3. However, it differs in that it has tried to draw a postscript instead of having intention of a user pushing a button in drawing 3 in now.

[0049] It is assumed that the user of a host computer draws a box on a computer. (This box is greatly illustrated too much for emphasis.) As for a box, it is desirable that it is inelastic over the computer itself. This invention reproduces the box on a remote computer. (This box is drawn using the postscript tool which is not illustrated).

[0050] Seeing by drawing 15, an input router sends the flow of logic to a "postscript." A "postscript" calls the suitable GDI function for drawing the box. A "postscript" sends a "postscript message" to connection API. This connection API sends a postscript message to a remote person.

[0051] The "postscript" shown in drawing 15 receives this postscript message. A this "postscript" block expresses the logic performed in each remote computer. This "postscript" calls a suitable GDI function via block GDI.

[0052] Graphical ["GDI"] Device It is a cable address of an interface. The "GDI function" is included in the big program called GDI.EXE. If a GDI function is called, it will draw a specific graphical image like a circle, a box, or a text based on the input which a user inputs after that. Other GDI functions perform other tasks, such as selection of pen width.

[0053] GDIEXE is a product marketed. "Programming of window drawing 3 .1" (Redmond, Washington.; Microsoft) of a "Windows software development kit" available from Microsoft Corp. in the technical details about GDI.EXE, and the Charles PETOZORUDO work [and] It is indicated to issue in 3 or ISBN 1-55615-395-1992.

[0054] The drawing 5 host runs an application program.

[0055] The mode is "application."

[0056] A user input is performed with a host computer.

[0057] A user tries use of a computer.

[0058] A host user moves cursor on the key "3" of a computer, and clicks a mouse. GUI generates a mouse message and sets it at cue. This invention calls a mouse message and sends the message to an application program (namely, computer program). This program carries out that the (1) key "3" was pressed and the response of drawing the number "3" on the display of a computer using the call of (2) GDI.

[0059] This application program records again the fact that the user inputted "3" in order to perform internal operation of that program.

[0060] When having drawn "3" in a computer, and when this invention displays the button of "3" pushed, it interrupts the GDI call which an application program issues. Other computers reproduce a host display by performing the same GDI function. The paragraph after entitling "general consideration" describes the explanation detailed one layer about this GDI discontinuation.

[0061] Thus, all the users can see a host's user operate a computer simultaneously. (Since very short delay involves, this host's operation is not strictly instantaneous.) But even if a host and a remote computer are located in a line and it operates, probably people cannot

notice the delay. An input router recognizes that an application program should point to a mouse message in drawing 15. It points to the flow of logic in application (namely, computer program). application draws the key of "3" (1) pushed -- (2) -- a suitable GDI function is called -- especially therefore, the number "3" is written in in the display of a computer.

[0062] However, this invention catches the GDI call of an application program via GDI capture shown in drawing 15, before they are performed (capture). This invention is attached to the caught call (call), and performs 2 things. As one of them, this invention gives notice of these calls to other computers via a "connection API" block. This action reaches the "GDI display caught" block shown in drawing 15. When it does so, this block makes each remote person perform the same GDI function so that block GDI may show.

[0063] As the 2, this invention allows performing the GDI function which the application program which should be executed by a host calls by a host via block GDI shown in drawing 15.

[0064] So, this invention catches the call (call) of the GDI function which the application program performed. As a result of this invention's giving notice of the caught call to a remote computer, the remote computer can reproduce those calls. This invention is allowed to perform on a host, as the caught call was meant.

[0065] The drawing 6 host runs an application program.

[0066] The mode is "a local postscript."

[0067] A user input is performed with a host computer.

[0068] It tries that a user makes it calculate to a computer.

[0069] In postscript mode, I will assume that there is no default postscript tool given to the user. Under this assumption, a user moves cursor even to the button of a computer and "it pushing" and the "input router" of drawing 15 pass a mouse message for a button to a "postscript" block. This mouse click is that which is not an effective postscript input sequence and which is not then (no tools are chosen), and does not draw "postscript" what.

[0070] A remote computer does not show movement of the cursor corresponding to the mouse of a host computer. It is because the line 5 of drawing 15 does not send a postscript message to other computers during execution of "a local postscript" as mentioned above.

[0071] Even if it pushes the button of a computer on a host display, it does not carry out redrawing the button answered and pushed on it again. It is because application did not receive a mouse message. It is drawing of the pushed computer button that application has responsibility.

[0072] When the default of the postscript is specified as the user who has taken local postscript mode, drawing by the tool is started by the user's mouse click. A user will end drawing by a publicly known method, when a user notices an error.

[0073] The drawing 7 host runs an application program.

[0074] The mode is "a local postscript."

[0075] A user input is performed with a host computer.

[0076] A user tries a postscript to a computer.

[0077] The input router shown in the A section of drawing 15 under these conditions recognizes effective postscripts, such as drawing of a box. An input router points to the flow of the logic in a "postscript" block, and calls the suitable GDI function for postscripts, such as drawing etc. of the box which a "postscript" block shows to drawing 7.

[0078] However, since the postscript is local, as shown in drawing 7, no box is drawn on a remote computer. No data is sent in accordance with the data path 5 of drawing 15.

[0079] The drawing 8 host runs an application program.

[0080] The mode is "observation."

[0081] A user input is performed with a host computer.

[0082] A user tries the operation of a computer.

[0083] As drawing 8 shows, the click of a mouse is disregarded, and nothing occurs in a remote computer.

[0084] Although an input router reads this mouse message in drawing 15, it is prevented so that application may not be reached. It is because the present mode is "observation mode (View Mode)."

[0085] The drawing 9 host runs an application program.

[0086] The mode is a "postscript."

[0087] A remote computer performs a user input.

[0088] A user tries the operation of a computer.

[0089] A user moves a mouse cursor on the button of a computer, and it is assumed that a mouse is clicked. This click is disregarded. Although other computers (a host and other

remote computers) show movement of a user's cursor, nothing shows them but it. This is because no tools are chosen.

[0090] In drawing 15, an input router prevents that this mouse message reaches application. A logical operation is applied to a "postscript" and a "postscript" draws cursor on a user's remote display via a GDI block. A "postscript" sends data to connection API and this connection API applies a logical operation to the "postscript" of drawing 15. this "postscript" expresses the contents of postscript logic which exist on other two computers (namely, a host -- and -- being concerned -- others -- a remote computer). A these "postscript" block draws at each place via the GDI block which shows drawing 15 the cursor corresponding to each user's cursor. GDI of drawing 15 expresses the contents of the GDI function call.

[0091] While a host can use one tool like a box draughting tool, the remote computer can use another tool like the tool describing a circle.

[0092] The drawing 10 host runs an application program.

[0093] The mode is a "postscript."

[0094] A remote computer performs a user input.

[0095] A user tries the postscript to a computer.

[0096] It is assumed that a postscript is a box. A box is drawn on all the displays. In drawing 15, the input router in a user's remote computer sends a mouse message to a "postscript" block. A "postscript" performs two things. One of them calls suitable GDI for a "postscript" to draw the box which is a postscript.

[0097] The 2, a "postscript" sends a postscript message to connection API, and connection API sends a postscript message to the computer of another side. However, these one side is host computers and another side is a remote computer. A host's logic process reaches the "postscript" of drawing 15, and the "postscript" which has a logic process of the remote computer of said another side in drawing 15 is reached.

[0098] Both call a suitable GDI function and a these "postscript" block makes the postscript of a user desire draw on this. However, a host's logical path 5 is not taken up at this time. It is because it is not necessary to reproduce a host's postscript to other computers.

[0099] The drawing 11 host runs an application program.

[0100] The mode is application.

[0101] A remote computer performs a user input.

[0102] A user tries the operation of a computer.

[0103] Readers need to recollect that this computer program is loaded only to the host. However, it hopes that the user of a remote computer wants to operate [it].

[0104] The input router of the remote user who shows drawing 15 sends a mouse message to connection API. A host receives these messages and these messages are sent to the input router of the host who shows drawing 15. As for a host's input router, block application sends these messages (to namely, application program which is a computer program). This program performs two important things.

[0105] A remote mouse -- them -- in spite of having carried out message appearance, the computer program treats those messages as if it was published with the host's mouse. A computer program will answer by the usual method, if it gets to know that the user inputted data (namely, "3"). The response should show the computer button "3" : [Containing the following thing] (1) Pushed. (2) Write in the number "3" in the display of a computer. And perform internal calculation of (3) itself.

[0106] However, before a computer program can perform now (1) of the foregoing paragraph, and (2), this invention catches the GDI function which a computer program calls first. This prehension is illustrated by the GDI prehension block of drawing 15. In the case of this prehension, this invention performs two things. This invention sends one of them of these GDI functions to connection API (in order to be able to use other computers). In a user's remote computer, connection API shown in drawing 15 sends this GDI function to "the caught GDI display." This display creates the duplicate of a host display. This invention performs the GDI function which should be performed by a host (going via block GDI shown in drawing 15) by a host as the 2. So, the general sequence of a phenomenon carries out the trial with which : and the remote user who is as follows push a computer button.

[0107] This invention which is operating on a remote computer detects this trial, and sends data to the computer program which is running on the host. This data takes the form of a message. The program of a computer "thinks" that this message came from the host's mouse.

[0108] A computer program usually works to a passage and draws an image on a host display via a GDI call.

[0109] This invention catches a GDI call and notify them to a remote computer.

[0110] A remote computer reproduces a host's window. The remote user can do in this way and can operate remotely the computer program which is running on the host.

[0111] If it summarizes by another way of speaking, this invention will generate a mouse message to a host based on the mouse message of a remote computer. (It is running on

the host.) A computer program answers a mouse message as if those mouse messages were generated in the host. The GDI call which the computer program performed is interrupted for this invention, and the same GDI call in a remote computer is performed by it.

Therefore, a host display is reproduced to a remote computer.

[0112] The drawing 12 host runs an application program.

[0113] The mode is "a local postscript."

[0114] A remote computer performs a user input.

[0115] A user tries OPERESSHON of a computer.

[0116] The mouse click of the user concerned is disregarded. Since selection of a tool has gone wrong, movement of this mouse is answered and nothing appears on other displays.

[0117] The drawing 13 host runs an application program.

[0118] The mode is "a local postscript."

[0119] A user input is performed by the remote computer.

[0120] A user tries the postscript to a computer.

[0121] As described above, a postscript is drawn on a user's display. No postscript occurs on other displays.

[0122] The drawing 14 host runs an application program.

[0123] The mode is "observation."

[0124] A remote computer performs a user input.

[0125] A user tries the operation of a computer.

[0126] A mouse click is disregarded although the cursor of a mouse moves on a user's display. Other two displays do not show movement of a user's mouse cursor.

[0127] The program of general consideration 1. another ** draws in a different portion of the whole display. A display is drawn using a GDI function. However, the portion from which at least one display differs is drawn by a different program, respectively.

[0128] In spite of performing these drawing [all] operations using a GDI function, GDI functions are not the exclusive display duplicate communication media between computers.

[0129] A postscript includes the data transfer between computers of one gestalten.

[0130] Drawing by an application program contains the thing of other gestalten.

[0131] For example, when a user puts a postscript, a user's mouse message is reproduced as a "message" in other computers via the course 5 shown in drawing 15. These duplicate messages rank second and make a suitable GDI call for an each (it can set to other computers) "postscript" block to draw the postscript publish. That is, a GDI call is not directly sent to other computers from the user who is performing the postscript.

[0132] When the application program is making the graphical image draw on a display, this invention interrupts the GDI call (going via "GDI prehension" shown in drawing 15), and makes the GDI call reproduce on other computers in contrast with it.

[0133] The display information of the main reasons for performing a different procedure (the duplicate of a mouse message and the duplicate of a GDI call) of two reasons for a difference is because a postscript is stored in a different location.

[0134] That is, it returns to the computer of drawing 2 and an application program stores the image of a computer by the following general methods. Postscript data is stored by this invention. Application program data is stored by an application program (a host has). The image of each key is stored as data in which a GDI function can pull out the key. This data includes information, including a position, a size, a color, etc. Each key contains the number of relation. The number is storable as a text character with information, including a position, a size, font types, etc.

[0135] Although postscript data is stored in a different place, it is stored by the same general method.

[0136] When any of a postscript or an application program they are needs a bit map, a bit map is stored by GUI by a publicly known method.

[0137] This invention combines a postscript image with the image of application by the publicly known art about masking. That is, this invention reproduces the GDI function which the remote computer received to a bit map. This invention reproduces the received postscript information to a different bit map from it. Mask credit of these two bit maps is carried out mutually.

[0138] For the purpose of a user enabling it to save an application image without a postscript etc., it dissociates from application data and postscript data is saved. As this substitute, the user can save postscript data in the form of a display put a postscript independently.

[0139] Another reason for carrying out separate preservation is convenient [saving postscript data separately] to draw the display which does not have postscript data at all. When postscript data has mixed with computer image data, it is difficult to remove postscript data also until it is not impossible.

[0140] Supposing it transmits a GDI call exclusively, and reproduction of a message is not performed at all namely,, excessive work with being separate for constituting so that it may store will be needed in postscript data.

[0141] 2. He can understand discontinuation of GDI, or discontinuation of prehension GDI as follows.

[0142] A. At the time of start-up, this invention replaces the first 5 bytes of each GDI function by a certain specific program (JUMP), i.e., the jump instructions to Trap.GDI.

[0143] B. Trap.GDI gains the parameter to a desired graphical image (it is a position of the angle of 2 diagonal lines in the case of a box, etc.). And PkgDispCall which is a sub program is called. Trap.GDI also replaces 5 bytes of the beginning again.

[0144] C. Receive a parameter from PkgDispCall and Trap.GDI and generate a BUJIEKUTO structure (object structure). This object structure is the data block included all the elements required for other computers to draw a box.

[0145] For example, the object structure includes the information about the size and position of a box. GUI draws an image in "context (context)." This context contains the thing of the pen which has others width and a color, and characteristic. This invention pursues the context of each computer. If the context of the drawn box differs from the context of the remote computer, PkgDispCall will include data required for other computers to generate a right context.

[0146] D. an object structure is shipped by other computers -- being concerned -- others - - perform the GDI function that a computer is the same.

[0147] E. This invention performs an original GDI function.

[0148] 3. The whole display is not transmitted. A display is not replaced by bitwise. For example, the image of the computer shown in drawing 2 can also transmit between computers by bitwise. If a computer occupies 200x300-pixel space, the information about 60,000 pixels (namely, pixel of 200x300) must be sent. As this substitute, the specific computer shown in drawing 2 is treated as 18 rectangles and 1 text sentence character to 18 rectangles of each, and, as a result, becomes 34 objects in all. Each object needs a size and a parameter like a position. The number of parameters is small and is 3 thru/or 10. 340 data must be sent if it assumes that there are ten parameters. Of course, although it is dependent on various factors, if a small number of bytes is assumed to each peace, it is sufficient for the size of each data piece (data pieces).

[0149] So, this invention reduces required data to a maximum of 340 peace for an object duplicate 60,000 piece with the duplicate of bitwise. Of course depending on an object, the gestalt of a bit map may be taken, and it must send for every bit. However, generally a bit map does not appear rarely. When sending a bit map generally, it is expected that it is sent only once.

[0150] Furthermore, object data is possible and **** compression is carried out. That is, signal transduction between computers will be performed in the form of compressed data, if possible. Compression is publicly known at this technical field.

[0151] 4. The communication between the gestalt computers of a data link can take some gestalten. The network, local-area, and wide area which are marketed can be used. The ISDN telephone line marketed is provided from the local telecommunications company, and can also use these. Modem communication can also be used.

[0152] 5. The package of marketing which detects the message which answered the message sensing input device of the advanced technology, and was generated in GUI can be used. One of such the packages is WINSIGHT, and this is Borland. It can obtain from International. However, these packages are considered not to carry out a message report at a remote computer.

[0153] 6. "GDI prehension" which becomes instead of
The method of replacing with prehension of the above-mentioned graphics is as follows. Before calling actual system GDI for GDI which this system gives, it replaces by the separate procedure which processes a GDI call. The name of system GDI is changed in order to prevent mixing up by two inter modules. In order to catch the GDI call performed via the module with which the system is provided, the same technique also as USRE.EXE is used.

[0154] 7. -- even -- the above computer can run an application program. A certain specific computer can act as a host to one program, and a remote computer can be acted to another program. For example, one computer can run a word processor and another computer can run a paint program. To each program, each is a host.

[0155] Since the software of this invention carried in each computer is the same or the same in a parenchyma top, it can run, as all the users mentioned above the word processor or the CAD program.

[0156] 8. There is "cursor" of the type of "real" cursor and two "false" cursor. Each GUI owns the "real (it real(s) and is actual)" cursor of itself. It is not generated that it is a GUI function but actual cursor is generated with the independent function in GUI. It can be considered that this cursor is a bit map to which you answer movement of a mouse and GUI makes it move. Adding to the real cursor controlled by a local mouse, this invention generates the "false" cursor to the participant in each remoteness. A pseudo-cursor is generated using a GDI function. Real cursor changes with movement of cursor occasionally. For example, when it is on a tool bar, the cursor can take the shape of an

arrow, and when it is on a client area, it can change to the form of a hand. This change is often set on the basis of control of an application program.

[0157] So, when a remote user (it is shown in drawing 11 like) controls the application program which is running on the host machine, an application program can change the cursor on a host machine, without using a GDI call. Therefore, prehension of GDI shown in drawing 15 and 15A is inefficiency to reproducing the cursor which changed on a remote display.

[0158] In order to cope with this problem, this invention supervises the function to change real cursor (for example, the SetCursor command etc.). This invention reproduces cursor change on a remote computer.

[0159] It is also one method to execute the same SetCursor command. The method instead of another will be changing remote cursor, when host cursor's changes, or making a bit map draw by performing the sequence of a suitable GDI call.

[0160] 9. The whole display is not reproduced. This invention only reproduces the window which the user of a display identifies. That is, the user can hold a piece of drawing like Note Pad in secret in a period of a meeting. A GDI call uses the task handle (taskhandle). A GDI call will not be shared if a task handle does not refer to the item to share.

[0161] 10. Always necessarily share no parameters of a GDI call.

[0162] Probably this feature of this invention can be best illustrated based on the following "backgrounds."

[0163] Issue of a background GDI call will evoke two main features. They are a "device context" and an "object." A device context is considered first.

[0164] Probably, the rectangle comprises the white line of a certain width drawn on the black background, supposing it draws one rectangle on a device context blackboard. This situation has illustrated the "device context" of the mold which consists of a white line of a certain line width provided on the black background. Any pictures drawn on the blackboard (using the same chalk) coincide with this device context.

[0165] In order to draw a rectangle on the output device of a computer like a video display, a printer, or a plotter, the same idea is applied when using GUI. We specify clearly the device context which one want to draw by default.

[0166] :by which the following parameter is contained in the device context which GUI uses -- a rectangular color and -- the color of a background, and -- the thickness of four lines which constitute a rectangle, and -- the size of the window in which a rectangle is accommodated, and a position -- and -- other data.

[0167] (GUI of desirable this example uses the device context which accommodates about 19 different parameters.)

A concept similar also to an object object is applied. A "pen" is chosen when publishing GDI for drawing a rectangle. This pen is a set of the parameter which describes not a physical pen but the line which should be drawn. The parameters of this pen are the molds (a solid line, a dashed line, a center line, etc.) of the width of a line, a color, and dashing (sketch). These parameters are stored by computer, and when calling in a pen in action, they are used.

[0168] Thus, GUI obtains drawing instructions from two or more source. The (1) GDI call which has the following three source in the example considered here. this -- the mold (rectangle) of a picture -- in addition to this, a size and a parameter like a position are specified.

[0169] (2) Device context. This is specified by GDI (it is by default and is used) call.

[0170] (3) The object which involves. This is also specified by GDI (not being by default) call.

[0171] When this invention relays a GDI call to a remote computer (relay), the device context and object which involve are what, or the remote computer must receive a notice. This invention compresses the information to transmit in order to protect prescribed band width.

[0172] I will assume the thing of ****.

[0173] (1) The teleconferencing session began now exactly.

[0174] (2) Application is shared.

[0175] (3) The GDI call is not published before now.

[0176] (4) Neither a device context nor a pen is corrected from the default.

[0177] (5) The application currently shared has appeared in the window named APP_WINNDOW.

[0178] (6) He expects that the user who is in a host computer draws a circle on "APP_WINNDOW."

[0179] If those who put a postscript draw a circle using a mouse, this invention will receive a mouse message, as mentioned above, and will publish a suitable GDI call. Clearly [a GDI call] about both a device context and a pen, by default, it can be referred to and, as a result, GUI can draw a circle now. by assumption, there is no change in the default value of a device context and a pen in any way -- since there is no ****, a default

value is used for GUI drawing a circle.

[0180] Since the application concerned is shared, this invention interrupts a GDI call on the way which calls GUI, and notifies the thing of the drawn circle to a remote computer. This invention must notify (b) device context and the information about the (c) pen to a remote computer with the identity of (a) GDI call. At this time, other data which is not important for this invention is combined, and it is notified.

[0181] there is no change in the default value of a device context and a pen in any way -- since there is no ****, notice of this invention is actually given to a remote computer so that both a default device context and a default pen may be used. This invention does not carry out transmitting the whole default device context, and does not carry out transmitting the whole default pen, either.

[0182] I will assume that those [other] who do an example postscript change either a device context or ****. For example, what (so, the parameter of a pen is changed) a postscript person changes the color of a background (therefore, a device context is changed), or the line width of a pen is changed for is made.

[0183] This invention does not carry out transmitting the whole device context to a remote computer, and does not carry out transmitting the whole set of a pen parameter, either. Instead, this invention transmits only the change concerned.

[0184] This invention transmits only the change concerned, only when the GDI call which performs the reference to the changed device context or the changed pen is published. An example in case this delay becomes important will be shown. It is assumed that the preparations to which a certain postscript person puts a postscript are made, and the color of the background of a window is changed, therefore the device context of the window concerned is changed during [another] the example teleconferencing session. It is assumed that the declaration which has a certain conference participant immediately after the change is performed. This declaration attracts a postscript person's attention and, for this reason, a postscript just before being carried out is interrupted.

[0185] Since [two] it is main, this invention does not transmit change of the device context concerned at this time. In the first place, it did not need to approach. It is because the GDI call which needs a device context is not published. It is better to use it rather than it for other transactions (processing), although such transmission will occupy a zone in the second. In this example, the above-mentioned declaration can sign the start of a lot of data transmission.

[0186] Instead, this invention stands by until a postscript person makes the GDI call which uses the changed device context publish. When the stage comes, this invention relays change to a remote person at the time. However, although said repeatedly again, the whole device context is not transmitted at this time, either. Only change is transmitted.

[0187] Another example remote computer stores both the device context in connection with the GDI call concerned, and the object. If a remote computer receives change, a remote person will correct suitably the device context and object which are stored. Therefore, a remote computer maintains the device context and object corresponding to the same device context and object which are stored in the host computer. A host updates the device context concerned and object if needed with a remote person. When not updated at all, a device context and an object maintain those default values.

[0188] If the drawing host computer which a remote person does not use the device context, but uses a host's GDI sends GDI to a remote person, A remote computer draws the picture requested by the GDI call using the already accommodated information into the device context of the self which carries out oneself correspondence, and the object. That is, a host transmits neither the whole device context required for the specified call, nor the whole object.

[0189] This aspect of affairs of this invention has especially a meaning, when a remote computer participates in the meeting which has already advanced. The host computer is maintaining a device context in use and the present status of an object. However, the remote computer which participates newly does not know the present status.

[0190] Nevertheless, since this invention transmits information to the remote person, the remote person can build a device context and an object from the information. In short, this information is produced change. A remote person builds a device context and an object based on the known default contents and the change concerned.

[0191] The above-mentioned argument considered the postscript. However, to the remote computer which does not put a postscript, this invention publishes a GDI call and transmits it. In these examples, :remote person who operates as this invention was mentioned above maintains a device context and an object based on change received from the host.

[0192] A definition "device context" can be defined as the set of the data which can be changed which specifies the background of the picture to draw. The example of such the data aggregate is indicated in the 514 thru/or 515-page table of the book of said PETOZORUDO.

[0193] Without transmitting the whole device context concerned based on change produced in the host's device context, The method of maintaining a device context in a remote computer can be called the "un-redundant" maintaining method ("non-redandant" maintenance).

[0194] That is, a host's device context has accommodated 19 parameters and will assume that the single parameter was changed. Probably, it will be redundant to transmit all 19 parameters to a remote person after change. It is because the remote person already knows 18 parameter values.

[0195] notes -- in desirable this example, if a change is made to an object like a pen, not only the change concerned but the whole object will be transmitted to other computers. This is because the data space which the whole object generally occupies is small.

[0196] [Effect of the Invention] Since a user can operate a computer program remotely, the electronic meeting system by which this invention was improved can operate the single program which exists in one of two or more of the computers.

[0197] in addition, the user of two or more computers is common to this invention -- display common use can be carried out. That is, a postscript can be put to a common display. As for the starting data communications to a remote computer, the device context concerning change and the video conference system which can carry out high speed communication since it ends with a little data communications since what is necessary is just to carry out object information transmission, and data can be compressed and sent can be given.

TECHNICAL FIELD

[Industrial Application] This invention relates to the system which can operate a computer program with two or more single users by remote control especially about a computer conference system.

PRIOR ART

[Description of the Prior Art] A modernistic phone system enables two or more parties which are in a different place to hold a conference. However, the conference call can give no facilities obtained at the meeting in which all participants meet on the common table of a conference room.

[0003] For example, in a conference room, the participant can see a drawing or a product in connection with subject, etc. Such observation is impossible in a conference call.

[0004] This invention relates to the system for performing common video conferencing using a phone system. This invention generates the common visual image distributed to some computers. The user of the computer concerned can put a postscript to the common picture (annotations). If a user performs this, this invention will reproduce the postscript on all the displays of all the computers.

[0005] This invention reduces the data volume distributed about generating of a common visual image.

EFFECT OF THE INVENTION

[Effect of the Invention] Since a user can operate a computer program remotely, the electronic meeting system by which this invention was improved can operate the single program which exists in one of two or more of the computers.

[0197] In addition, the user of two or more computers is common to this invention -- display common use can be carried out. That is, a postscript can be put to a common display. As for the starting data communications to a remote computer, the device context concerning change and the video conference system which can carry out high speed communication since it ends with a little data communications since what is necessary is just to carry out object information transmission, and data can be compressed and sent can be given.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] The technical problem of this invention is giving the improved electronic meeting system.

[0007] Another technical problem of this invention is giving the system by which it enables it for a user to operate a computer program remotely.

[0008] Another technical problem of this invention is giving the system by which it enables it for those computers to operate the single program which exists in one of two or more of the computers.

[0009] Another technical problem of this invention is giving the system to which it can be added while the user of two or more computers can see a common display.

[0010] Another technical problem of this invention is giving the video conference system which reduces the data volume transmitted between computers.

MEANS

[Means for Solving the Problem] This invention creates two or more duplicates of the video image concerned by transmitting a command for generating a common video image. Required data volume of these commands is smaller than a bit map on transmission.

[0012] However, in order to draw the picture concerned, a program which is running on a computer needs big data rather than contained in the command itself. Data which should be added is supplied using a data structure accommodated in a computer. These data structures are often called a "device context" or an "object", and specify a parameter like a color of a line, or width of a line. A user can change a data structure, and a data structure is maintained by default value when a user does nothing.

[0013] If a data structure is changed, this invention does not carry out notifying the change to other computers of all the immediately, but it will wait for it until it is necessary for other computers to get to know about the change.

EXAMPLE

[Example] General-view drawing 1 shows three computers connected with the telephone wire. Each computer runs the multitasking graphical user interface (Graphical User Interface, GUI) by which a message drive is carried out. The example has a thing named the Windows marketed from Microsoft Corporation of Redmond, Washington. It is such. GUI is also called operating environment (operating environments) again.

[0015] the user of GUI gives an interaction via the program using Windows -- it can do. This invention reproduces not the whole display of the computer which exists far away but the window of shoes to choose. A user becomes possible [maintaining individual area on its own display (one's area which is not shared)] by this alternative duplicate. Each computer can run the software which this invention person developed again. One computer (host) runs an application program. (Since it has the multitasking ability of GUI, a host It is possible to run both programs).

[0016] This invention has the operation of four dominant modes.

[0017] 1. Application mode (Application Mode) Any user of three computers shown in drawing 1 can give a command to an application program. For example, an application program will assume that it is what simulates a stock type computer. The first state is shown in drawing 2. In this case, each computer display illustrates a computer. The user of : and the host who assumes that the following phenomenon happens pushes the button "3" of a computer (it responds to the design of a computer program and they are an input of a keyboard, or an input of a mouse). This is answered and each computer displays "3" on the viewing area.

[0018] - One of remote users pushes "+." Other remote users push "6."

[0019] - A host user pushes "=",

[0020] At this time, all the computers display "9" which is the sum of 3 and 6. Users perform operation of this computer program in a group, and each one of displays display that result.

[0021] This computer program does not take into consideration whether it did not take into consideration whether which user pushed the button, and someone of users pushed the button, as long as a lawful sequence is received. (However, it is assumed that users are cooperative and neither of the users does nonfeasance of the operation of a computer).

[0022]2. Postscript mode (Annotation Mode) Any user can draw on a user's own local

display using the same drawing tool as what is seen by the "paint" program. The user can draw a box, a circle, a circle, an ellipse, etc. The user can also eliminate the item on a display.

[0023] This invention can reproduce users' postscript on other displays of all the, therefore can see a display with all the similar users. However, since there are the following factors, those displays may differ.

[0024] (A) If display monitors differ, character, such as expression ability of a color, differs.

[0025] (B) If the protocols (IGA, BGA, etc.) of a display differ, expressions of a graphical image differ and display ability differs.

[0026] (C) GUI differs. Or it is the same. When GUI also differs in a version, display types may differ.

[0027] (D) Some users might change the size of the window where a computer is displayed, and, as a result, the difference may have produced them in the contraction scale.

[0028] These differences make the appearance of the image displayed produce a difference. However, the fundamental contents of all the displays are the same ****. Since the difference in a size is permitted, this invention can be drawn by suitable contraction scale.

[0029] 3. Local postscript mode (Local Annotation Mode) Although the user can put a postscript to a local display, since those postscripts are kept secret, other users cannot see the postscript.

[0030] 4. Observation mode (View Mode) Neither of the users can put a postscript and a command cannot be published. However, action similar to a postscript can be taken. Since the user can move his cursor and other users can see the motion, remote indication (indication of the position of a display thing with cursor) can be performed. Observation mode is useful in the example etc. which specify the mode of the specific user who is, for example during execution in postscript mode as observation mode. In this example, although all the users can put a postscript, "observation" user can only see and cannot put a postscript.

[0031] Explanatory view 3 in each mode - drawing 14 illustrates various modes through use of a computer program as an example.

[0032] The drawing 3 host runs an application program.

[0033] The mode is a "postscript."

[0034] A user's input is performed with a host computer.

[0035] A user tries the operation of a computer.

[0036] It is assumed that the user of a host computer tries addition of two numbers using a computer. The first number that is the target of the trial is considered.

[0037] A host user moves a host's cursor on the key of a computer, and a keystroke is tried by clicking a mouse. However, the click of a mouse does not reach an application program. It is because this invention prevents it. An application program does not answer. It is because a program does not receive the click of a mouse.

[0038] That is, if it explains still in detail, GUI will detect movement of this mouse and will generate the "mouse message" which should be generated. GUI sets this mouse message at cue (queue). A mouse message stands by processing in this cue. "The input router (INPUT ROUTER)" shown in drawing 15 reads these messages. Since "postscript mode" is performing now, an input router sends this message to a postscript block. Application does not receive this message, therefore does not answer. This mouse click is disregarded.

[0039] The response "postscript" of a "postscript" can be constituted so that two kinds (or more than it) of responses may be carried out to the message of a mouse. In the one composition, a postscript requires that a mouse should choose a postscript tool first. If selection is not performed at all, a "postscript" will disregard a mouse message.

[0040] Selection is performed by clicking a mouse on a tool image so that it may usually be carried out by a "paint" program. A "postscript" recognizes selection of this tool and, subsequently treats the click of the mouse after it as data for drawing with a selection tool. For example, when a rectangle tool is chosen, the click of the following two mice becomes final and conclusive the corner on a rectangular diagonal line. (Drawing 4 about which it argues later illustrates rectangular drawing.)

In the second composition, when it is in postscript mode, a default tool like a pen is chosen automatically. With this composition, if a user performs the trial which pushes the button of a computer (it clicks), drawing of the line of the user using that pen will start (** which is not meant). The user who has recognized this can make drawing of a line finish using either of the publicly known methods.

[0041] So, postscript In the mode, since the (b "postscript") tool is not answered or chosen as the input of a mouse by starting the "postscript" of the (a) default, this invention carries out the response of ***** which disregards the input of a mouse. The keyboard input from a user is treated similarly. It can also design so that a "postscript" may carry out other responses, of course.

[0042] Pursuit each display of cursor shows the cursor which can be spotted with the mouse of relation. This invention can reproduce each cursor on all the displays. Thus, as shown in drawing 3, when using 3 mice, there is three cursor. (However, since it is easy, even free accepts it and it has illustrated it.)

As a result, if a one user moves a mouse, the cursor corresponding to it will exercise on all the displays.

[0043] Generally :each cursor which three cursor can identify identifies the owner by introducing for example, a color, shape, and a label etc.

[0044] The drawing 4 host runs an application program.

[0045] The mode is postscript mode.

[0046] A user's input is performed with a host computer.

[0047] A user tries drawing of a box on a computer.

[0048] This situation is extremely similar with what is shown in drawing 3. However, it differs in that it has tried to draw a postscript instead of having intention of a user pushing a button in drawing 3 in now.

[0049] It is assumed that the user of a host computer draws a box on a computer. (This box is greatly illustrated too much for emphasis.) As for a box, it is desirable that it is inelastic over the computer itself. This invention reproduces the box on a remote computer. (This box is drawn using the postscript tool which is not illustrated).

[0050] Seeing by drawing 15, an input router sends the flow of logic to a "postscript." A "postscript" calls the suitable GDI function for drawing the box. A "postscript" sends a "postscript message" to connection API. This connection API sends a postscript message to a remote person.

[0051] The "postscript" shown in drawing 15 receives this postscript message. A this "postscript" block expresses the logic performed in each remote computer. This "postscript" calls a suitable GDI function via block GDI.

[0052] Graphical ["GDI"] Device It is a cable address of an interface. The "GDI function" is included in the big program called GDI.EXE. If a GDI function is called, it will draw a specific graphical image like a circle, a box, or a text based on the input which a user inputs after that. Other GDI functions perform other tasks, such as selection of pen width.

[0053] GDIEXE is a product marketed. "Programming of window drawing 3 .1" (Redmond, Washington.; Microsoft) of a "Windows software development kit" available from Microsoft Corp. in the technical details about GDI.EXE, and the Charles

PETOSORUDO work [and] It is indicated to issue in 3 or ISBN 1-55615-395-1992.

[0054] The drawing 5 host runs an application program.

[0055] The mode is "application."

[0056] A user input is performed with a host computer.

[0057] A user tries use of a computer.

[0058] A host user moves cursor on the key "3" of a computer, and clicks a mouse. GUI generates a mouse message and sets it at cue. This invention calls a mouse message and sends the message to an application program (namely, computer program). This program carries out that the (1) key "3" was pressed and the response of drawing the number "3" on the display of a computer using the call of (2) GDI.

[0059] This application program records again the fact that the user inputted "3" in order to perform internal operation of that program.

[0060] When having drawn "3" in a computer, and when this invention displays the button of "3" pushed, it interrupts the GDI call which an application program issues. Other computers reproduce a host display by performing the same GDI function. The paragraph after entitling "general consideration" describes the explanation detailed one layer about this GDI discontinuation.

[0061] Thus, all the users can see a host's user operate a computer simultaneously. (Since very short delay involves, this host's operation is not strictly instantaneous.) But even if a host and a remote computer are located in a line and it operates, probably people cannot notice the delay. An input router recognizes that an application program should point to a mouse message in drawing 15. It points to the flow of logic in application (namely, computer program). application draws the key of "3" (1) pushed -- (2) -- a suitable GDI function is called -- especially therefore, the number "3" is written in in the display of a computer.

[0062] However, this invention catches the GDI call of an application program via GDI capture shown in drawing 15, before they are performed (capture). This invention is attached to the caught call (call), and performs 2 things. As one of them, this invention gives notice of these calls to other computers via a "connection API" block. This action reaches the "GDI display caught" block shown in drawing 15. When it does so, this block makes each remote person perform the same GDI function so that block GDI may show.

[0063] As the 2, this invention allows performing the GDI function which the application program which should be executed by a host calls by a host via block GDI shown in drawing 15.

[0064] So, this invention catches the call (call) of the GDI function which the application program performed. As a result of this invention's giving notice of the caught call to a remote computer, the remote computer can reproduce those calls. This invention is allowed to perform on a host, as the caught call was meant.

[0065] The drawing 6 host runs an application program.

[0066] The mode is "a local postscript."

[0067] A user input is performed with a host computer.

[0068] It tries that a user makes it calculate to a computer.

[0069] In postscript mode, I will assume that there is no default postscript tool given to the user. Under this assumption, a user moves cursor even to the button of a computer and "it pushing" and the "input router" of drawing 15 pass a mouse message for a button to a "postscript" block. This mouse click is that which is not an effective postscript input sequence and which is not then (no tools are chosen), and does not draw "postscript" what.

[0070] A remote computer does not show movement of the cursor corresponding to the mouse of a host computer. It is because the line 5 of drawing 15 does not send a postscript message to other computers during execution of "a local postscript" as mentioned above.

[0071] Even if it pushes the button of a computer on a host display, it does not carry out redrawing the button answered and pushed on it again. It is because application did not receive a mouse message. It is drawing of the pushed computer button that application has responsibility.

[0072] When the default of the postscript is specified as the user who has taken local postscript mode, drawing by the tool is started by the user's mouse click. A user will end drawing by a publicly known method, when a user notices an error.

[0073] The drawing 7 host runs an application program.

[0074] The mode is "a local postscript."

[0075] A user input is performed with a host computer.

[0076] A user tries a postscript to a computer.

[0077] The input router shown in the A section of drawing 15 under these conditions recognizes effective postscripts, such as drawing of a box. An input router points to the flow of the logic in a "postscript" block, and calls the suitable GDI function for

postscripts, such as drawing etc. of the box which a "postscript" block shows to drawing 7.

[0078] However, since the postscript is local, as shown in drawing 7, no box is drawn on a remote computer. No data is sent in accordance with the data path 5 of drawing 15.

[0079] The drawing 8 host runs an application program.

[0080] The mode is "observation."

[0081] A user input is performed with a host computer.

[0082] A user tries the operation of a computer.

[0083] As drawing 8 shows, the click of a mouse is disregarded, and nothing occurs in a remote computer.

[0084] Although an input router reads this mouse message in drawing 15, it is prevented so that application may not be reached. It is because the present mode is "observation mode (View Mode)."

[0085] The drawing 9 host runs an application program.

[0086] The mode is a "postscript."

[0087] A remote computer performs a user input.

[0088] A user tries the operation of a computer.

[0089] A user moves a mouse cursor on the button of a computer, and it is assumed that a mouse is clicked. This click is disregarded. Although other computers (a host and other remote computers) show movement of a user's cursor, nothing shows them but it. This is because no tools are chosen.

[0090] In drawing 15, an input router prevents that this mouse message reaches application. A logical operation is applied to a "postscript" and a "postscript" draws cursor on a user's remote display via a GDI block. A "postscript" sends data to connection API and this connection API applies a logical operation to the "postscript" of drawing 15. this "postscript" expresses the contents of postscript logic which exist on other two computers (namely, a host -- and -- being concerned -- others -- a remote computer). A these "postscript" block draws at each place via the GDI block which shows drawing 15 the cursor corresponding to each user's cursor. GDI of drawing 15 expresses the contents of the GDI function call.

[0091] While a host can use one tool like a box draughting tool, the remote computer can use another tool like the tool describing a circle.

[0092] The drawing 10 host runs an application program.

[0093] The mode is a "postscript."

[0094] A remote computer performs a user input.

[0095] A user tries the postscript to a computer.

[0096] It is assumed that a postscript is a box. A box is drawn on all the displays. In drawing 15, the input router in a user's remote computer sends a mouse message to a "postscript" block. A "postscript" performs two things. One of them calls suitable GDI for a "postscript" to draw the box which is a postscript.

[0097] The 2, a "postscript" sends a postscript message to connection API, and connection API sends a postscript message to the computer of another side. However, these one side is host computers and another side is a remote computer. A host's logic process reaches the "postscript" of drawing 15, and the "postscript" which has a logic process of the remote computer of said another side in drawing 15 is reached.

[0098] Both call a suitable GDI function and a these "postscript" block makes the postscript of a user desire draw on this. However, a host's logical path 5 is not taken up at this time. It is because it is not necessary to reproduce a host's postscript to other computers.

[0099] The drawing 11 host runs an application program.

[0100] The mode is application.

[0101] A remote computer performs a user input.

[0102] A user tries the operation of a computer.

[0103] Readers need to recollect that this computer program is loaded only to the host. However, it hopes that the user of a remote computer wants to operate [it].

[0104] The input router of the remote user who shows drawing 15 sends a mouse message to connection API. A host receives these messages and these messages are sent to the input router of the host who shows drawing 15. As for a host's input router, block application sends these messages (to namely, application program which is a computer program). This program performs two important things.

[0105] a remote mouse -- them -- in spite of having carried out message appearance, the computer program treats those messages as if it was published with the host's mouse. A computer program will answer by the usual method, if it gets to know that the user inputted data (namely, "3"). The response should show the computer button "3" : [

Containing the following thing] (1) Pushed. (2) Write in the number "3" in the display of a computer. And perform internal calculation of (3) itself.

[0106] However, before a computer program can perform now (1) of the foregoing paragraph, and (2), this invention catches the GDI function which a computer program calls first. This prehension is illustrated by the GDI prehension block of drawing 15. In the case of this prehension, this invention performs two things. This invention sends one of them of these GDI functions to connection API (in order to be able to use other computers). In a user's remote computer, connection API shown in drawing 15 sends this GDI function to "the caught GDI display." This display creates the duplicate of a host display. This invention performs the GDI function which should be performed by a host (going via block GDI shown in drawing 15) by a host as the 2. So, the general sequence of a phenomenon carries out the trial with which : and the remote user who is as follows push a computer button.

[0107] - This invention which is operating on a remote computer detects this trial, and sends data to the computer program which is running on the host. This data takes the form of a message. The program of a computer "thinks" that this message came from the host's mouse.

[0108] - A computer program usually works to a passage and draws an image on a host display via a GDI call.

[0109] - This invention catches a GDI call and notify them to a remote computer.

[0110] - A remote computer reproduces a host's window. The remote user can do in this way and can operate remotely the computer program which is running on the host.

[0111] If it summarizes by another way of speaking, this invention will generate a mouse message to a host based on the mouse message of a remote computer. (It is running on the host) A computer program answers a mouse message as if those mouse messages were generated in the host. The GDI call which the computer program performed is interrupted for this invention, and the same GDI call in a remote computer is performed by it.

Therefore, a host display is reproduced to a remote computer.

[0112] The drawing 12 host runs an application program.

[0113] The mode is "a local postscript."

[0114] A remote computer performs a user input.

[0115] A user tries OPERESSHON of a computer.

[0116] The mouse click of the user concerned is disregarded. Since selection of a tool has gone wrong, movement of this mouse is answered and nothing appears on other displays.

[0117] The drawing 13 host runs an application program.

[0118] The mode is "a local postscript."

[0119] A user input is performed by the remote computer.

[0120] A user tries the postscript to a computer.

[0121] As described above, a postscript is drawn on a user's display. No postscript occurs on other displays.

[0122] The drawing 14 host runs an application program.

[0123] The mode is "observation."

[0124] A remote computer performs a user input.

[0125] A user tries the operation of a computer.

[0126] A mouse click is disregarded although the cursor of a mouse moves on a user's display. Other two displays do not show movement of a user's mouse cursor.

[0127] The program of general consideration 1. another ** draws in a different portion of the whole display. A display is drawn using a GDI function. However, the portion from which at least one display differs is drawn by a different program, respectively.

[0128] In spite of performing these drawing [all] operations using a GDI function, GDI functions are not the exclusive display duplicate communication media between computers.

[0129] A postscript includes the data transfer between computers of one gestalt.

[0130] Drawing by an application program contains the thing of other gestalten.

[0131] For example, when a user puts a postscript, a user's mouse message is reproduced as a "message" in other computers via the course 5 shown in drawing 15. These duplicate messages rank second and make a suitable GDI call for an each (it can set to other computers) "postscript" block to draw the postscript publish. That is, a GDI call is not directly sent to other computers from the user who is performing the postscript.

[0132] When the application program is making the graphical image draw on a display, this invention interrupts the GDI call (going via "GDI prehension" shown in drawing 15), and makes the GDI call reproduce on other computers in contrast with it.

[0133] The display information of the main reasons for performing a different procedure (the duplicate of a mouse message and the duplicate of a GDI call) of two reasons for a difference is because a postscript is stored in a different location.

[0134] That is, it returns to the computer of drawing 2 and an application program stores the image of a computer by the following general methods. Postscript data is stored by this invention. Application program data is stored by an application program (a host has). The image of each key is stored as data in which a GDI function can pull out the key. This data includes information, including a position, a size, a color, etc. Each key contains the number of relation. The number is storable as a text character with information, including a position, a size, font types, etc.

[0135] Although postscript data is stored in a different place, it is stored by the same general method.

[0136] When any of a postscript or an application program they are needs a bit map, a bit map is stored by GUI by a publicly known method.

[0137] This invention combines a postscript image with the image of application by the publicly known art about masking. That is, this invention reproduces the GDI function which the remote computer received to a bit map. This invention reproduces the received postscript information to a different bit map from it. Mask credit of these two bit maps is carried out mutually.

[0138] For the purpose of a user enabling it to save an application image without a postscript etc., it dissociates from application data and postscript data is saved. As this substitute, the user can save postscript data in the form of a display put a postscript independently.

[0139] Another reason for carrying out separate preservation is convenient [saving postscript data separately] to draw the display which does not have postscript data at all. When postscript data has mixed with computer image data, it is difficult to remove postscript data also until it is not impossible.

[0140] supposing it transmits a GDI call exclusively, and reproduction of a message is not performed at all namely,, excessive work with being separate for constituting so that it may store will be needed in postscript data.

[0141] 2. He can understand discontinuation of GDI, or discontinuation of prehension GDI as follows.

[0142] A. At the time of start-up, this invention replaces the first 5 bytes of each GDI function by a certain specific program (JUMP), i.e., the jump instructions to Trap.GDI.

[0143] B. Trap.GDI gains the parameter to a desired graphical image (it is a position of the angle of 2 diagonal lines in the case of a box, etc.). And PkgDispCall which is a sub

program is called. Trap.GDI also replaces 5 bytes of the beginning again.

[0144] C. Receive a parameter from PkgDispCall and Trap.GDI and generate a BUJIEKUTO structure (object structure). This object structure is the data block included all the elements required for other computers to draw a box.

[0145] For example, the object structure includes the information about the size and position of a box. GUI draws an image in "context (context)." This context contains the thing of the pen which has others width and a color, and characteristic. This invention pursues the context of each computer. If the context of the drawn box differs from the context of the remote computer, PkgDispCall will include data required for other computers to generate a right context.

[0146] D. an object structure is shipped by other computers -- being concerned -- others - - perform the GDI function that a computer is the same.

[0147] E. This invention performs an original GDI function.

[0148] 3. The whole display is not transmitted. A display is not replaced by bitwise. For example, the image of the computer shown in drawing 2 can also transmit between computers by bitwise. If a computer occupies 200x300-pixel space, the information about 60,000 pixels (namely, pixel of 200x300) must be sent. As this substitute, the specific computer shown in drawing 2 is treated as 18 rectangles and 1 text sentence character to 18 rectangles of each, and, as a result, becomes 34 objects in all. Each object needs a size and a parameter like a position. The number of parameters is small and is 3 thru/or 10. 340 data must be sent if it assumes that there are ten parameters. Of course, although it is dependent on various factors, if a small number of bytes is assumed to each peace, it is sufficient for the size of each data piece (data pieces).

[0149] So, this invention reduces required data to a maximum of 340 peace for an object duplicate 60,000 piece with the duplicate of bitwise. Of course depending on an object, the gestalt of a bit map may be taken, and it must send for every bit. However, generally a bit map does not appear rarely. When sending a bit map generally, it is expected that it is sent only once.

[0150] Furthermore, object data is possible and **** compression is carried out. That is, signal transduction between computers will be performed in the form of compressed data, if possible. Compression is publicly known at this technical field.

[0151] 4. The communication between the gestalt computers of a data link can take some gestalten. The network, local-area, and wide area which are marketed can be used. The ISDN telephone line marketed is provided from the local telecommunications company, and can also use these. Modem communication can also be used.

[0152] 5. The package of marketing which detects the message which answered the message sensing input device of the advanced technology, and was generated in GUI can

be used. One of such the packages is WINSIGHT, and this is Borland. It can obtain from International. However, these packages are considered not to carry out a message report at a remote computer.

[0153] 6. "GDI prehension" which becomes instead of The method of replacing with prehension of the above-mentioned graphics is as follows. Before calling actual system GDI for GDI which this system gives, it replaces by the separate procedure which processes a GDI call. The name of system GDI is changed in order to prevent mixing up by two inter modules. In order to catch the GDI call performed via the module with which the system is provided, the same technique also as `USR.EXE` is used.

[0154] 7. -- even -- the above computer can run an application program. A certain specific computer can act as a host to one program, and a remote computer can be acted to another program. For example, one computer can run a word processor and another computer can run a paint program. To each program, each is a host.

[0155] Since the software of this invention carried in each computer is the same or the same in a parenchyma top, it can run, as all the users mentioned above the word processor or the CAD program.

[0156] 8. There is "cursor" of the type of "real" cursor and two "false" cursor. Each GUI owns the "real (it real(s) and is actual)" cursor of itself. It is not generated that it is a GUI function but actual cursor is generated with the independent function in GUI. It can be considered that this cursor is a bit map to which you answer movement of a mouse and GUI makes it move. Adding to the real cursor controlled by a local mouse, this invention generates the "false" cursor to the participant in each remoteness. A pseudo-cursor is generated using a GDI function. Real cursor changes with movement of cursor occasionally. For example, when it is on a tool bar, the cursor can take the shape of an arrow, and when it is on a client area, it can change to the form of a hand. This change is often set on the basis of control of an application program.

[0157] So, when a remote user (it is shown in drawing 11 like) controls the application program which is running on the host machine, an application program can change the cursor on a host machine, without using a GDI call. Therefore, prehension of GDI shown in drawing 15 and 15A is inefficiency to reproducing the cursor which changed on a remote display.

[0158] In order to cope with this problem, this invention supervises the function to change real cursor (for example, the `SetCursor` command etc.). This invention reproduces cursor change on a remote computer.

[0159] It is also one method to execute the same `SetCursor` command. The method instead of another will be changing remote cursor, when host cursor's changes, or making a bit map draw by performing the sequence of a suitable GDI call.

[0160] 9. The whole display is not reproduced. This invention only reproduces the window which the user of a display identifies. That is, the user can hold a piece of drawing like Note Pad in secret in a period of a meeting. A GDI call uses the task handle (taskhandle). A GDI call will not be shared if a task handle does not refer to the item to share.

[0161] 10. Always necessarily share no parameters of a GDI call.

[0162] Probably this feature of this invention can be best illustrated based on the following "backgrounds."

[0163] Issue of a background GDI call will evoke two main features. They are a "device context" and an "object." A device context is considered first.

[0164] Probably, the rectangle comprises the white line of a certain width drawn on the black background, supposing it draws one rectangle on a device context blackboard. This situation has illustrated the "device context" of the mold which consists of a white line of a certain line width provided on the black background. Any pictures drawn on the blackboard (using the same chalk) coincide with this device context.

[0165] In order to draw a rectangle on the output device of a computer like a video display, a printer, or a plotter, the same idea is applied when using GUI. We specify clearly the device context which one want to draw by default.

[0166] : by which the following parameter is contained in the device context which GUI uses -- a rectangular color and -- the color of a background, and -- the thickness of four lines which constitute a rectangle, and -- the size of the window in which a rectangle is accommodated, and a position -- and -- other data.

[0167] (GUI of desirable this example uses the device context which accommodates about 19 different parameters.) A concept similar also to an object object is applied. A "pen" is chosen when publishing GDI for drawing a rectangle. This pen is a set of the parameter which describes not a physical pen but the line which should be drawn. The parameters of this pen are the molds (a solid line, a dashed line, a center line, etc.) of the width of a line, a color, and dashing (sketch). These parameters are stored by computer, and when calling in a pen in action, they are used.

[0168] Thus, GUI obtains drawing instructions from two or more source. The (1) GDI call which has the following three source in the example considered here. this -- the mold (rectangle) of a picture -- in addition to this, a size and a parameter like a position are specified.

[0169] (2) Device context. This is specified by GDI (it is by default and is a call).

[0170] (3) The object which involves. This is also specified by GDI (not being by default) call.

[0171] When this invention relays a GDI call to a remote computer (relay), the device context and object which involve are what, or the remote computer must receive a notice. This invention compresses the information to transmit in order to protect prescribed band width.

[0172] I will assume the thing of ****.

[0173] (1) The teleconferencing session began now exactly.

[0174] (2) Application is shared.

[0175] (3) The GDI call is not published before now.

[0176] (4) Neither a device context nor a pen is corrected from the default.

[0177] (5) The application currently shared has appeared in the window named APP_WINNDOW.

[0178] (6) He expects that the user who is in a host computer draws a circle on "APP_WINNDOW."

[0179] If those who put a postscript draw a circle using a mouse, this invention will receive a mouse message, as mentioned above, and will publish a suitable GDI call. Clearly [a GDI call] about both a device context and a pen, by default, it can be referred to and, as a result, GUI can draw a circle now. by assumption, there is no change in the default value of a device context and a pen in any way -- since there is no ****, a default value is used for GUI drawing a circle.

[0180] Since the application concerned is shared, this invention interrupts a GDI call on the way which calls GUI, and notifies the thing of the drawn circle to a remote computer. This invention must notify (b) device context and the information about the (c) pen to a remote computer with the identity of (a) GDI call. At this time, other data which is not important for this invention is combined, and it is notified.

[0181] there is no change in the default value of a device context and a pen in any way -- since there is no ****, notice of this invention is actually given to a remote computer so that both a default device context and a default pen may be used. This invention does not carry out transmitting the whole default device context, and does not carry out transmitting the whole default pen, either.

[0182] I will assume that those [other] who do an example postscript change either a device context or ****. For example, what (so, the parameter of a pen is changed) a postscript person changes the color of a background (therefore, a device context is

changed), or the line width of a pen is changed for is made.

[0183] This invention does not carry out transmitting the whole device context to a remote computer, and does not carry out transmitting the whole set of a pen parameter, either. Instead, this invention transmits only the change concerned.

[0184] This invention transmits only the change concerned, only when the GDI call which performs the reference to the changed device context or the changed pen is published. An example in case this delay becomes important will be shown.

It is assumed that the preparations to which a certain postscript person puts a postscript are made, and the color of the background of a window is changed, therefore the device context of the window concerned is changed during [another] the example teleconferencing session. It is assumed that the declaration which has a certain conference participant immediately after the change is performed. This declaration attracts a postscript person's attention and, for this reason, a postscript just before being carried out is interrupted.

[0185] Since [two] it is main, this invention does not transmit change of the device context concerned at this time. In the first place, it did not need to approach. It is because the GDI call which needs a device context is not published. It is better to use it rather than it for other transactions (processing), although such transmission will occupy a zone in the second. In this example, the above-mentioned declaration can sign the start of a lot of data transmission.

[0186] Instead, this invention stands by until a postscript person makes the GDI call which uses the changed device context publish. When the stage comes, this invention relays change to a remote person at the time. However, although said repeatedly again, the whole device context is not transmitted at this time, either. Only change is transmitted.

[0187] Another example remote computer stores both the device context in connection with the GDI call concerned, and the object. If a remote computer receives change, a remote person will correct suitably the device context and object which are stored. Therefore, a remote computer maintains the device context and object corresponding to the same device context and object which are stored in the host computer. A host updates the device context concerned and object if needed with a remote person. When not updated at all, a device context and an object maintain those default values.

[0188] If the drawing host computer which a remote person does not use the device context, but uses a host's GDI sends GDI to a remote person, A remote computer draws the picture requested by the GDI call using the already accommodated information into the device context of the self which carries out oneself correspondence, and the object. That is, a host transmits neither the whole device context required for the specified call, nor the whole object.

[0189] This aspect of affairs of this invention has especially a meaning, when a remote computer participates in the meeting which has already advanced. The host computer is maintaining a device context in use and the present status of an object. However, the remote computer which participates newly does not know the present status.

[0190] Nevertheless, since this invention transmits information to the remote person, the remote person can build a device context and an object from the information. In short, this information is produced change. A remote person builds a device context and an object based on the known default contents and the change concerned.

[0191] The above-mentioned argument considered the postscript. However, to the remote computer which does not put a postscript, this invention publishes a GDI call and transmits it. In these examples, :remote person who operates as this invention was mentioned above maintains a device context and an object based on change received from the host.

[0192] A definition "device context" can be defined as the set of the data which can be changed which specifies the background of the picture to draw. The example of such the data aggregate is indicated in the 514 thru/or 515-page table of the book of said PETOZORUDO.

[0193] Without transmitting the whole device context concerned based on change produced in the host's device context, The method of maintaining a device context in a remote computer can be called the "un-redundant" maintaining method ("non-redandant" maintenance).

[0194] That is, a host's device context has accommodated 19 parameters and will assume that the single parameter was changed. Probably, it will be redundant to transmit all 19 parameters to a remote person after change. It is because the remote person already knows 18 parameter values.

[0195] notes -- in desirable this example, if a change is made to an object like a pen, not only the change concerned but the whole object will be transmitted to other computers.

This is because the data space which the whole object generally occupies is small.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1]It is a figure which illustrates three computers connected with the telephone wire.

[Drawing 2]An example of an example explained in detail is shown in 3 thru/or drawing 14 of a figure. In this example, the computer program operates by various parties and can attach a postscript.

[Drawing 3]When this invention is in postscript mode and one of host users tries the operation of a computer, it is a figure showing how this invention answers.

[Drawing 4]When this invention is in postscript mode and one of host users tries a postscript to a computer, it is a figure showing how this invention answers.

[Drawing 5]When an application mode has this invention and one of host users tries the operation of a computer, it is a figure showing how this invention answers.

[Drawing 6]When this invention is in local postscript mode and one of host users tries the operation of a computer, it is a figure showing how this invention answers.

[Drawing 7]When this invention is in local postscript mode and one of host users tries a postscript to a computer, it is a figure showing how this invention answers.

[Drawing 8]When this invention is in a view mode, it is a figure showing how it answers in a host user.

[Drawing 9]When this invention is in postscript mode and a remote user tries the operation of a computer, it is a figure showing how this invention answers.

[Drawing 10]When this invention is in postscript mode and a remote user tries a postscript to a computer, it is a figure showing how this invention answers.

[Drawing 11]When this invention is in an application mode and a remote user tries the operation of a computer, it is a figure showing how this invention answers.

[Drawing 12]When this invention is in local postscript mode and a remote user tries the operation of a computer, it is a figure showing how this invention answers.

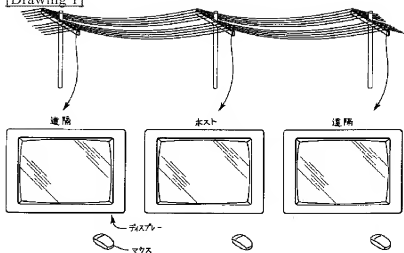
[Drawing 13]When this invention is in local postscript mode and a remote user tries a postscript to a computer, it is a figure showing how this invention answers.

[Drawing 14]When this invention is in a view mode, it is a figure showing how it answers to a remote user.

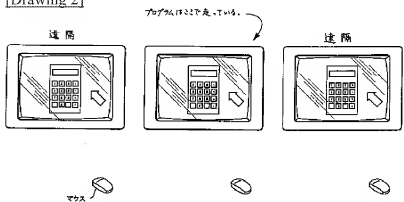
[Drawing 15]It is a figure which illustrates the flow of the logic used by this invention.

DRAWINGS

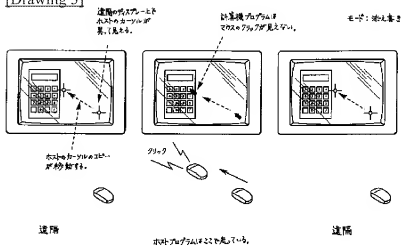
[Drawing 1]



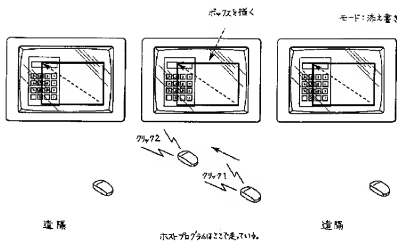
[Drawing 2]



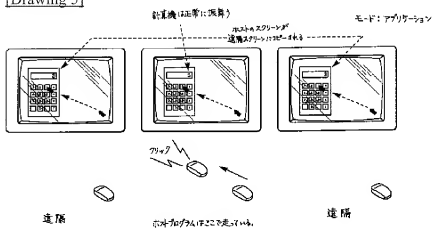
[Drawing 3]



[Drawing 4]

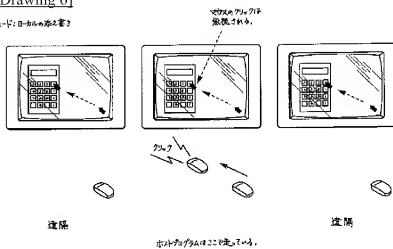


[Drawing 5]



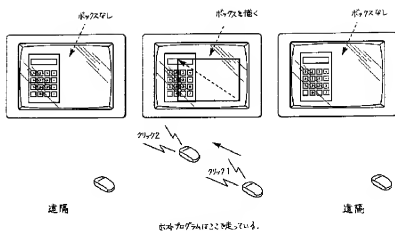
[Drawing 6]

モード: 目おろしの添え書き



[Drawing 7]

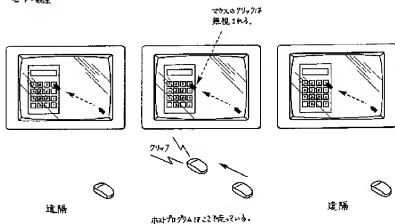
モード：ローカルの通話書き



ボタンスワレはここを押している。

[Drawing 8]

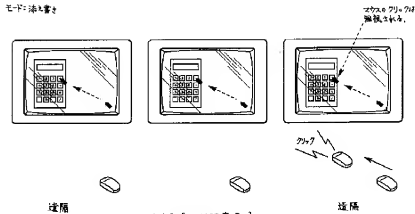
モード：録音



ボタンスワレはここを押している。

[Drawing 9]

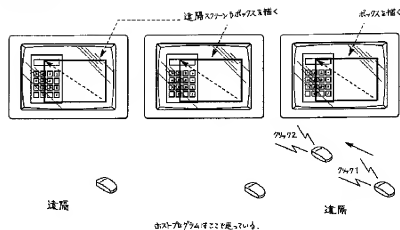
モード：通話書き



ボタンスワレはここを押している。

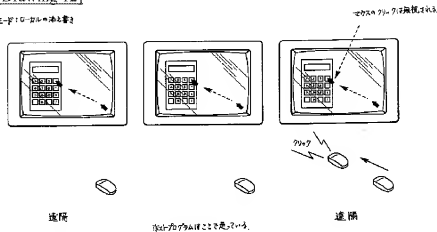
[Drawing 10]

モード: 通入書き



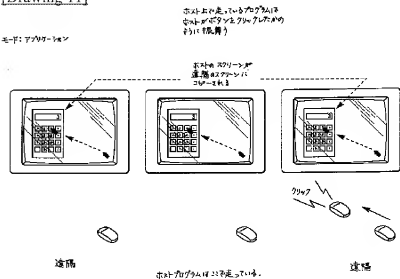
[Drawing 12]

モード: ローカルホスト書き



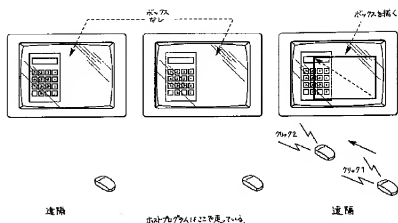
[Drawing 11]

モード: アンダーウェア



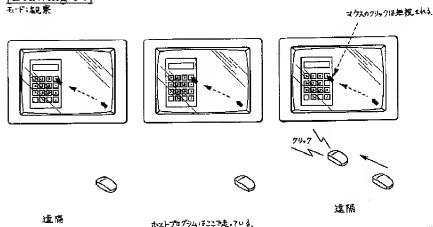
[Drawing 13]

モード：ローカルモード



[Drawing 14]

モード：監視



[Drawing 15]

